KYMENLAAKSON AMMATTIKORKEAKOULU
Viestintä / Digitaalinen media

Eveliina Korjus

# VERKKOSIVUN RAKENTAMINEN JA TIETOTURVA – FUNCTIONAL SAFETY FINLAND

Opinnäytetyö 2013

TIIVISTELMÄ

KYMENLAAKSON AMMATTIKORKEAKOULU
Viestintä

| | |
|---|---|
| KORJUS, EVELIINA | Verkkosivun rakentaminen ja tietoturva – Functional Safety Finland |
| Opinnäytetyö | 34 sivua + 7 liitesivua |
| Työn ohjaaja | Pt. tuntiopettaja Suvi Pylvänen |
| Toimeksiantaja | Functional Safety Finland |
| Huhtikuu 2013 | |
| Avainsanat | verkkosivut, tietoturva, käytettävyys, verkkosuunnittelu, sisällönhallintajärjestelmä, Silverstripe |

Työn tavoitteena on tutustua verkkosivujen ja eri koodikielten suurimpiin tietoturvariskeihin ja ongelmiin ja selvittää, miten ne saadaan korjattua mahdollisimman vähällä vaivalla. Työssä käydään myös läpi verkkosivustojen käytettävyysperiaatteita ja konventioita sekä pienimuotoisen käytettävyystestauksen perusteita, jolloin käyttäjälle voidaan luoda selkeä ja miellyttävä verkkosivusto, jolta käyttäjä löytää helposti tarvitsemansa informaation.

Produktiivisena työnä rakennetaan uudet verkkosivut kotkalaiselle Functional Safety Finlandille, joka tarjoaa toiminnallisen turvallisuuden koulutus- ja elinkaaripalveluita toisille yrityksille. FSF:n vanhat sivut olivat kankeat, vanhanaikaiset ja sisällöltään yritystä jäljessä. Graafikko Sofia Herkiä oli luonut FSF:lle uuden, selkeän ja raikkaan yritysilmeen juuri ennen projektin alkua, joten sivuston rakenteeseen pystytään keskittymään paremmin.

Ensimmäiseksi keskitytään tietoturvaongelmiin, käydään läpi tietoturvan yleiset periaatteet ja merkityksen sekä suunnittelijan että loppukäyttäjän kannalta ja varmistetaan työn tietoturvan sopivimmalla tavalla. Sen jälkeen tutkitaan Internetissä vallitsevia konventioita ja rakennetaan niiden pohjalta mahdollisimman selkeän sivuston. Jo näin lyhyt tutustuminen sekä tietoturvaan että konventioihin on ollut valaiseva kokemus, joka on opettanut enemmän kuin aluksi uskoin.

ABSTRACT

KYMENLAAKSON AMMATTIKORKEAKOULU
University of Applied Sciences
Media Communications

| | |
|---|---|
| KORJUS, EVELIINA | Web development and information security - Functional Safety Finland |
| Bachelor's Thesis | 34 pages + 7 pages of appendices |
| Supervisor | Suvi Pylvänen, lecturer |
| Commissioned by | Functional Safety Finland |
| March 2013 | |
| Keywords | websites, information security, usability, web design, content management system, Silverstripe |

The objective of my thesis is to get familiarise with the most common information security issues and risks in web development and in different coding languages, and examine the simplest methods of repairing them. It will also go through the general rules of website usability and long-lasting conventions of the Internet today, so that the designer can provide the user with the clearest and the most pleasant possible web experience and to make sure the user will be able to find the information he requires.

The productive project consists of building a new and improved website for Functional Safety Finland, a Kotka-based company providing functional safety training and life cycle services for other businesses. The previous website the company maintained was out of date, unwieldy and lacking in content. Graphic designer Sofia Herkiä had dedicated her thesis for creating a new and fresh corporate identity for Functional Safety Finland, thus enabling me to focus on the structure of the site.

First the thesis will go through web security issues and the general principles of information security from both the designer's and the user's perspective, and verify the security of my own project in the best possible way. Afterwards it will use the common web conventions to build a clear, easy-to-use and functioning website. This brief survey of information concerning web security and design conventions has been a truly eye-opening experience of which I have learned more than I had anticipated.

SISÄLLYS

LIITTEET

# 1 JOHDANTO

Opinnäytetyöni tarkoituksena on rakentaa käytettävyydeltään, toiminnaltaan ja turvallisuudeltaan onnistuneet verkkosivut kotkalaiselle toiminnallisen turvallisuuden koulutusta ja palveluita tarjoavalle Functional Safety Finland -yritykselle. Yrityksen entiset sivut olivat jääneet pahasti ajastaan jälkeen eivätkä tarjonneet asiakkaan tarvitsemaa tietoa. Opinnäytetyössä käydään nopeasti läpi verkkosivun rakentamisen ja sivuston käytettävyyden perusteita. Työssä keskitytään verkkosivustojen tietoturva-asioihin niin suunnittelijan, käyttäjän kuin palveluidenkin kannalta. Käyn läpi sitä, miten suunnittelija voi suojella sivustoaan yleisimpiä tietoturvauhkia vastaan yksinkertaisin keinoin. Käyn myös läpi eri koodien tietoturva-aukoista ne, joihin kaikkien verkkosivujen rakentajien tulisi kiinnittää huomiota sivuston kokoon katsomatta. Lisäksi työssäni kerrotaan, mitä käyttäjältä tulisi olettaa ja vaatia, jotta varmistuttaisiin, että sivusto on turvallinen myös loppukäyttäjän käsissä.

Suurimpina haasteitani ovat täysin tuntematon kohdeala ja projektin laajuus. Minulla ei ollut etukäteen minkäänlaista käsitystä toiminnallisesta turvallisuudesta eikä sen vaatimuksista. Tämän takia kuitenkin pystyn pitämään mielessäni ne käyttäjät, joille ala on aivan yhtä vieras, vaikka sivusto suunnitellaankin alaa jo tunteville ja sen palveluita etsiville. En myöskään ennen ollut tutustunut tietoturva-asioihin tarkemmin, vaikka tietoturvariskit ovat jatkuva uhka verkottuneessa. Näin opinnäytetyöstä on hyötyä niin minulle itselleni kuin myös tuleville sivustojen rakentajille ja opinnäytetöiden tekijöille. Toivoisin, että tulevaisuudessa verkkosuunnittelijat ottavat heti aluksi huomioon edes yksinkertaisimmat tietoturvauhat ja varmistuisivat siitä, että ovat tietoisia, mihin riskiin itsensä ja sivunsa laittavat huolimattomalla koodillaan.

Yhteyshenkilönä toimi Functional Safety Finlandin perustaja ja toiminnallisen turvallisuuden asiantuntija Mikko Heikkilä. Functional Safety Finland oli jo suunnitellut uusia sivuja jonkin verran, joten sisällön sekä pääpiirteisen ulkoasun kehittäminen oli helppoa.

# 2 TIETOTURVA VERKKOSIVULLA

## 2.1 Mitä on tietoturva?

Usein kuvitellaan, että salaus eli tietojen muuttaminen salattuun muotoon ja tietoturva olisivat sama asia. Kuitenkin tietoturva on paljon laajempi käsite, jonka tavoitteena on suojella tietojenkäsittelyn prosessia ulkopuolisia uhkia vastaan joko salauksen avulla tai ilman. Usein tietoturvauhat johtuvat ihmisten tekojen seurauksena, mutta myös tekniset viat, kovalevyjen tuhoutumiset tai muut peruuttamattomat tapaturmat kuuluvat tietoturvan piiriin. (Järvinen 2003, 29.)

Salaukset eivät ole tietoturvan kulmakivi. Vaikka salaus estää tiedon päätymisen väärin käsiin, usein salauksen murtaminen on tietoa havittelevien krakkereiden eli haittaa ja omaa etuaan tavoittelevien tietomurtautujien (Järvinen 2004, 111) viimeinen keino. Salausten murtaminen on hidasta, vaivalloista ja turhaa, sillä on huomattavasti helpompaa saada käsiinsä salasanoja ja koodia esimerkiksi lähestymällä salasanan tietävää käyttäjää ja valehdellen manipuloida tämä luovuttamaan salasanansa. Tällaisessa tilanteessa tiedon salauksella ei ole enää mitään merkitystä.

Verkkosivuston sisällön ja tietojen turvaaminen on ensiarvoisen tärkeää, kun kyseessä oleva sivusto sisältää ulkopuolisten, rekisteröityneiden käyttäjien tietoja, erityisesti tässä tilanteessa, sillä kyseessä on luottamukseen ja turvallisuuteen perustuvia palveluita tarjoava ala. Vaikka tietokannassa ei olisikaan henkilö- ja pankkitietoihin verrattavaa informaatiota, olisi tietomurto vakava isku Functional Safety Finlandin imagolle. Yrityksen suhteellisen pienen koon vuoksi ei kohdistettu verkkohyökkäys ole todennäköinen, on palveluntarjoaja Radicenter suuremmassa vaarassa. Krakkerit käyttävät yksittäisten sivustojen haavoittuvuuksia päästäkseen käsiksi koko palveluntarjoajan tietokantaan. Ammattitaitoista krakkeria vastaan on liki mahdotonta puolustautua, mutta arkipäiväisempien hyökkäysyritysten torjuminen on helppoa. Käyn seuraavissa luvuissa läpi sitä, miten hyökkäysten torjuminen onnistuu.

## 2.2 Yleisimmät hyökkäystavat

*Cross-site scripting* tarkoittaa vieraan ja usein vihamielisen JavaScriptin lisäämistä ulkopuoliselle serverille muiden käyttäjien haitaksi, vaikkapa tekstikenttien avulla. Koodin avulla voidaan kaapata käyttäjän evästeet, napinpainallukset, salasanat ja sivuhistoria. Näillä tiedoilla selain voi tyhjentää pankkitilejä, kaapata muita sivustoja ja

käyttäjiä sekä vaikkapa julkaista herjaavia kommentteja verkkofoorumeilla. Cross-site scripting on yleisin verkon turvariskeistä ja oli vuonna 2007 vastuussa reilusta 80 % turvallisuusaukoista (Symantec 2007). Cross-site scriptingiä käytetään usein ohittamaan sivustojen saman alkuperän käytäntöjä, joilla estetään käytössä olevan, luotetun verkkosivuston ulkopuolisen koodin toiminta. Saman alkuperän käytäntö tuotettiin alun perin juuri cross-site scriptingiä vastaan 1990-luvun puolessavälissä, kun verkkosivustot koostuivat vielä taulukoista ja oli helppoa ladata ulkopuolisen koodin avulla uusi sivu tai linkki yhteen taulukoiden sarakkeista. Tällöin cross-site scripting oli vielä nimensä mukaista, mutta sen sisältö on parissa kymmenessä vuodessa muuttunut nykyiseen malliinsa (Grossman 2006).

Cross-site script -hyökkäysten välttämiseen on yksi nyrkkisääntö: käyttäjään ei saa luottaa. Kaikkeen käyttäjän syöttämään tietoon on suhtauduttava varauksella, ja jos mahdollista, tietojen tallentamista välimuistiin tulisi välttää, esimerkiksi tilanteessa, jossa käyttäjä epäonnistuu salasanan syöttämisessä ja sivusto tallentaa käyttäjätunnuksen, jottei käyttäjän tarvitsisi syöttää sitä uudestaan. Käyttäjätunnuksen syöttökenttä näyttäisi koodissa tältä:

*<input type="text" name="username" value="${ user_name }" />*

Tällaiseen koodiin on helppo syöttää käyttäjätunnukseksi esimerkiksi

*”><script>alert("hello")</script><input*

jolloin input-tunniste päätetään keinotekoisesti value-määreen sisään. Input-tunnisteen sisälle voidaan lisätä script-tunniste, jonka perään lisätään toinen input-tunnisteen loppu, merkillä <input />. Lopullinen koodi siis näyttäisi tältä (kuva 1):

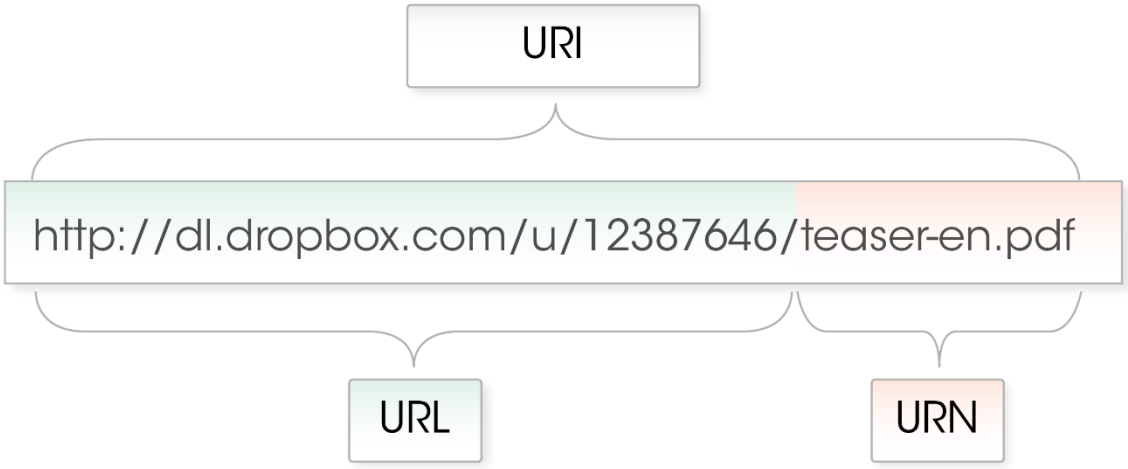
```
1 <input type="text" name="username"
  value=""><script>alert("hello") </script><input " />
```

Kuva 1. Yksinkertainen XSS-hyökkäys.

Tällöin kirjautumisnappia painamalla selain suorittaa koodin alert(”hello”), jolloin käyttäjä näkee hello-popup-ikkunan. Tämä ei tietenkään ole millään tavoin haitallista, joskin hieman ärsyttävää, mutta koodin voi lisätä mitä vain. Helpon tapa estää tämän kaltaiset hyökkäykset on lisätä input-tunnisteeseen value-määreeksi escape_special, joka estää erikoismerkkien käytön tekstikentässä. (Salihefendic 2009.)

<input type="text" name="username" value="${ escape_special(user_name) }" />

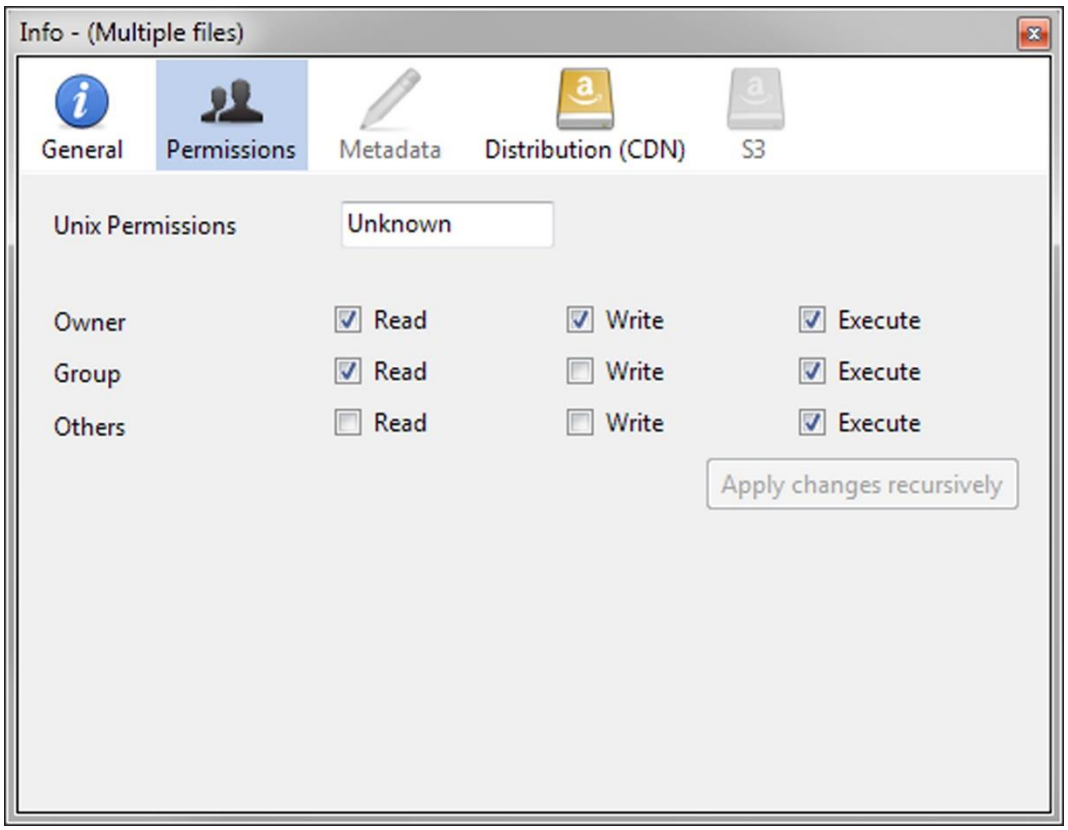
*URI eli uniform resource identifier* on uniform resource locator (URL) ja uniform resource name (URN) -käsitteiden kattotermi, ja on merkkijono, jota käytetään abstraktien tai fyysisten resurssien tunnistamiseen. URN määrittää kohteen identiteetin ja URL tavan, jolla kohde voidaan paikallistaa. URN voi olla vaikkapa koneella tai serverillä sijaitseva tiedostonimi, ja URL on siihen vievä tiedostopolku. (World Wide Web Consortium 2001.)




Kuva 2. Mistä kaikesta URI muodostuu.

URI sisältää usein myös muuttujia, jotka HTML-käyttöliittymä siihen liittää. Näitä ovat esimerkiksi hakujen yhteydessä URI:in lisätyt hakusanat ja -määreet, jolla selain osaa tietoa etsiä. Määreitä on kuitenkin melko helppoa kirjoittaa URI:in käsin. Jos esimerkiksi Ylen osoitteen perään kirjoittaa http://yle.fi/haku/default_fi.jsp?g1.qry=syyria&g1.oft=81, päädyttäisiin sivulle, jossa Yle  näyttää kaikki Syyriaan liittyvät uutiset alkaen 81. hakutuloksesta. Muuttujia voi syöttää myös sivuston elementtien kautta. Käytännössä kaikki elementit, joihin voi sisällyttää href- tai src-määreen, voidaan käyttää muuttujien syöttämiseen. Jos hakkeri pääsee muuttamaan HTML-tiedostojen elementtejä tai lisäämään omansa, voi hakkeri lisätä URI:in omat muuttujansa ja päästä näin käsiksi muualla serverillä sijaitseviin tiedostoihin. (Heilmann 2010.)

Puolustuskonstit URI-hyökkäystä vastaan ovat yksinkertaisia. On pyrittävä pitämään sivuston tiedostojen oikeudet mahdollisimman ankarina. Esimerkiksi kehityskansioon ei ole asiaa kenelläkään muulla kuin kehittäjällä ja omistajalla. Kaikissa ftp-ohjelmissa, kuten Cyberduck ja Filezilla, on sisäänrakennettu oikeuksienhallinta, jolla voidaan määritellä lukemis-, kirjoittamis- ja suoritusoikeudet. (Kuva 3.)




Kuva 3. CyberDuck-ohjelman oikeuksienhallintaikkuna. Satunnaiskäyttäjillä (others) ei ole oikeutta muuhun, kuin sivuston katselemiseen (execute).

*Path traversal* on hyökkäys, jossa hyökkääjä pyrkii pääsemään käsiksi verkkosivuston juurikansion, eli verkkosivuilla usein public_html-kansion, ulkopuolella sijaitseviin tiedostoihin. Tämä onnistuu absoluuttisten linkkien avulla, jotka hakevat käyttöönsä jonkin tiedoston, esimerkiksi

*http://some_site.com.br/get-files.jsp?file=report.pdf*

Hyökkääjä pystyy tämän URL:in avulla liikkumaan tiedostopuussa taaksepäin käyttämällä .../-sekvenssiä, joka on yleisesti käytössä CSS-koodissa, kun linkitetään aiemmissa kansioissa olleita tiedostoja, kuten kuvia. Jokainen ../-sekvenssi vie yhden kerroksen taaksepäin, ja jos sen perään kirjoitetaan siinä kerroksessa sijaitsevan toisen kansion nimi, päästään tiedostopuussa toiseen suuntaan.

*http://some_site.com.br/get-files?file=../../../../some dir/some file*

Helpoin tapa estää vieraiden pääsy käsiksi heille kuulumattomiin kansioihin ja tiedostoihin on sama kuin URI-hyökkäyksessä, eli pitää tiedostojen oikeudet mahdollisimman tiukkoina.

## 2.3 Tietoturva suunnittelijan kannalta

Yksinkertaisin tietoturvan parannus on kansiolistauksen estäminen (folder listing). Jos sivustolta puuttuu etusivuksi määritelty sivu, yleensä index-html, käyttäjän pyrkiessä sivustolle selain listaa kaiken sivutilalta löytyvän tiedon selkeästi puumalliin. Tämä on jo itsestään suuri tietoturva-aukko, mutta erityisesti silloin, jos henkilötietoja ei ole mitenkään salattu ja jos testaustarkoitukseen käytettäviä kansioita (esimerkiksi hiekkalaatikot) ei ole salasanasuojattu tai muutan piilotettu. Kansiolistauksen estäminen on lyhyt ja yksinkertainen temppu, joka onnistuu joko sivutilan hallintapaneelin (esimerkiksi cPanel) kautta tai suoraan serveriltä, sivuston juuresta löytyvän htaccess-tai httpd.conf-tiedoston kautta. cPanelista löytyy kohta nimeltä Index Manager, josta päästään hallinnoimaan kaikkien sivutilan kansioiden listauksia. Htaccess-tiedostoon vain lisätään teksti ”Options -Indexes” tai korvataan jo olemassa oleva options-kohta. Httpd.conf -tiedostoon, jota käytetään vain tietokoneen sisäisellä localhost-virtuaalipalvelimella, lisätään sama komento. (Techie Corner 2007.)

Vaikka salausmenetelmät eivät olekaan tietoturvassa niin tärkeitä kuin usein luullaan, on kannattavaa salata kaikki sivuston kautta kulkeva tärkeä tieto. Käyttäjätunnukset, salasanat, täytetyt kaavakkeet sekä lähetetyt tiedostot ja erityisesti rahaliikenne tulisi salata tavalla tai toisella, jos mahdollista käyttäen TLS- eli transport layer security -protokollaa (vanhemmalta nimeltään secure sockets layer SSL), jolloin palvelin todentaa itsensä käyttäjälle. TLS-varmenne on sidottu palvelimen domain- tai IP-osoitteeseen, jolloin käyttäjä voi olla varma palvelimen oikeellisuudesta. Jos TSL-protokolla on käytössä, näkyy käyttäjälle osoitteen edessä https://. Ainoana ongelmana on tilanne, jossa ulkopuolinen taho on päässyt fyysisesti käsiksi palvelimeen ja korvannut siellä sijaitsevan datan haluamallaan versiolla. (Järvinen 2003, 41.) SSL-sertifikaatin voi joissakin tilanteissa ostaa omalta verkkotilan tarjoajaltaan tai tilata ulkopuolisilta yrityksiltä.

### 2.3.1 Tiedon kryptaus

Jokaisen sivuston, jolle on mahdollista rekisteröityä, tulisi pitää tarkasti huolta käyttäjien salasanojen kryptaamisesta eli salauksesta. Kryptauksessa tieto muutetaan sellaiseen muotoon, jonka vain salausavaimen tunteva taho voi lukea. (Järvinen 2004, 193–194.) Vaikka aikaisemmin mainitsin, että salaus ei ole niin tärkeä osa tietoturvaa kuin ihmiset uskovat (ks. 4.1), salasanojen kohdalla se on suorastaan välttämätöntä. Koska rekisteröitynyt käyttäjä on luotetumpi kuin rekisteröimätön, pääsevät rekisteröityneet

käyttäjät usein käsiksi arkaluontoisempiin tietoihin, erityisesti sivuston hallinnoijan tai sisällönhallinnoitsijoiden tilanteessa. Salaamaton salasana on suhteellisen helposti saavutettavissa edellä mainituilla hyökkäystavoilla, ja jos salasanat ovat ihmisten luettavassa formaatissa, hakkeria ei estä mikään.

Salaus muuttaa salasanat hasheiksi eli merkkijonoiksi, jotka ovat nopeammin tietokannan käsiteltävissä. On saatavilla monia erilaisia algoritmejä, jotka muuntavat salasanat erilaisiksi hasheiksi. Esimerkiksi internetistä löytynyt MD5-hashgeneraattori muutti esimerkkisalasanan ”taikamuki” hashiksi ”15b04aa1c988a4d6eabec5ef1fe8f121”. Hasheja murretaan eri tavoin, joista suurimpina brute force ja rainbow table. Brute force (suomeksi väsytysmenetelmä) tarkoittaa yksinkertaisimmillaan sitä, että kone jätetään arvaamaan salasanan tarkoitusta yksi merkki kerrallaan (Järvinen 2003, 56). Arvausvauhdit ovat nykyään huimia. Nykyiset koneet voivat saavuttaa viiden miljardin arvauksen sekuntivauhdin. Rainbow table sen sijaan vertaa hasheja jo murrettuihin hash-tietokantoihin, joita on helppo ostaa internetin välityksellä.

Yleisimmin käytössä olevat algoritmit ovat jo murretut MD4 ja 5, turvallisena pidetty SHA-256, murtamaton Twofish sekä Rijndael, jonka Yhdysvaltain standardoimisviraston NIST valitsi vuonna 2001vielä murtamattoman AES:n (advanced encryption standard) pohjaksi, jota Yhdysvaltain hallitus nykyään käyttää. (Rouse 2011.) Eri algoritmit käyttävät erikokoisia avainkokoja ja usein tarjoavat käyttäjälleen mahdollisuuden muuttaa käytettävän avainkoon kokoa. Esimerkiksi SHA-256 käyttää nimensä mukaisesti 256 bitin mittaista hashia. En tässä tekstissä paljasta, minkä algoritmin valitsin, mutta suunnittelijan on hyvä tutkia algoritminsä taustat huolellisesti ja miettiä, onko algoritmi tarkoitukseen sopiva. Vaikka kovempi salaus on periaatteessa aina parempi, on harkittava, tarvitseeko juuri kyseinen sivusto hallitustasoista suojausta.

### 2.3.2 HTML- ja CSS-tietoturvariskit

Vaikka html (hypertext markup language) on suhteellisen turvallista jo pelkästään sen takia, että siihen harvemmin säilötään mitään valonarkaa tietoa, pitää html:n kanssa aina muistaa se tosiasia, että koodi on kaikkien ihmisten nähtävillä ja muuteltavissa. Myös piilotetut koodit, kommentit ja esimerkiksi monivalintakaavakkeiden vaihtoehdot näkyvät lähdekoodissa. Tämän vuoksi julkisiin html-dokumentteihin ei koskaan saisi sisällyttää keskeneräisiä osioita eikä niitä tulisi kommentoida. (Heilmann 2010.)

Lisäksi html:n on helppo sisällyttää muita koodikieliä. Sivuston käyttäjä voi lähettää esimerkiksi kaavakkeen avulla serverille html:n kätkettyä javascriptiä, jolla voidaan

saada vakaa vahinkoa aikaan. Script-tunnisteen (<*script*>) voi kiertää sisällyttämällä se esimerkiksi body-tunnisteeseen (<*body onload="..*">) tai div-tunnisteeseen laukaisukomennon kera (<*div onmouseover="myBadScript()"* />). Vaikka serveri tarkistaisi kaiken käyttäjän sinne lähettämän datan script-tunnisteiden varalta, jäävät html:n piilotetut skriptit siltä kokonaan huomaamatta. (Stack Overflow 2011.) Koska koodia voi piilottaa html:n monella eri vapaa, olisi hyvin työlästä rakentaa itse kaiken lähetetyn tiedon tarkistava koodi. Onneksi tarjolla on vapaasti saatavia html-koodivahteja, esimerkiksi HTML Purifier. Tietenkin internetistä ladattujen tarkistusohjelmien kanssa tulee olla tarkka ja asentaa vain luotettavista lähteistä saatuja puhdistajia.

CSS (cascading style sheets) on html:n tavoin itsessään turvallinen. Koska CSS määrittää vain sivuston ulkoasua, on koodi yksinkertaisimmillaan vesitiivis. Ongelmia syntyy vasta, kun CSS-tiedostoon linkitetään ulkopuolista dataa, esimerkiksi @font-face -määreellä. Tällöin CSS-tiedostoon liitetään linkki serverillä sijaitsevaan fonttitiedostoon, jota CSS käyttää sivuston näyttämiseen. Jos fonttitiedosto sijaitsee ulkopuolisella serverillä, se voidaan helposti vaihtaa fonttitiedoston kuosiin piilotettuun haitalliseen koodiin. Tämä sama vaara pätee kaikkiin tilanteisiin, joissa CSS-tiedostoon linkitetään tai tuodaan ulkopuolista dataa. (Heilmann 2010.)

### 2.3.3 PHP-tietoturvariskit

PHP:n jää helposti hyväksikäytettäviä tietoturva-aukkoja, jotka johtuvat usein huolimattomasta ja laiskasta koodista. Esimerkiksi globaalit muuttujat ovat helposti kenen tahansa asiaan vähänkään perehtyneen käytettävissä. Lisäksi PHP:n virheilmoitukset ovat krakkereille tehokas tiedon lähde. Virheilmoitukset voi estää lisäämällä PHP-tiedostoihin käskyn *error_reporting(0);*. Tämä kannattaa kuitenkin tehdä vasta julkaisuvaiheessa, sillä kehityksen ollessa kesken virheilmoitukset ovat tärkeitä, kunhan kansio itsessään on turvattu. (Heilmann 2010.) PHP:n turvallisuuden voi tarkastuttaa serverille asennettavalla PHPSecInfo-nimisellä työkalulla, joka listaa kaikki koodista löytämänsä turvaongelmat. Tietoja ei tietenkään pidä jättää serverille hakkereiden saataville.

Suurin PHP:n tietoturvariski on SQL-injektio, joka syöttää tietoja SQL-tietokantaan. Aivan kuten cross-site scriptingissäkin hyökkäysväylänä toimivat syöttötiedot, joiden oikeellisuutta ei ole todennettu mitenkään. Tässä tapauksessa kuitenkin hyökkäyksen kohteena on itse tietokanta, joka voidaan tuhota, lukita tai lähettää eteenpäin. Tieto-

kannat sisältävät niin kaiken sivuston ja ohjelmien tarvitseman tiedon kuin myös esimerkiksi rekisteröityneiden käyttäjien henkilökohtaisia tietoja. Suurin osa hyökkäyksistä voidaan torjua samoin kuin cross-site scripting eli estämällä erikoismerkkien käyttö. (PHP 2013.) Lisäksi PHP:hen voidaan kirjata yksinkertainen mysql_real_escape_string()-funktio, joka tarkistaa syötetyn tiedon käyttökelpoisuuden SQL-tietokannan kannalta. Esimerkissä näytetään, miten opiskelijan nimen syöttö tietokantaan toimii (kuva 4).

```
1  <?php
2  $db = mysql_connect('localhost', 'username', 'password');
3  mysql_select_db('school', $db);
4  $studentName = mysql_real_escape_string($_POST[
   'student_name'], $db);
5  $queryResult = mysql_query("INSERT INTO Students (name)
   VALUE ('{$studentName}')");
6  if ($queryResult) {
7          echo 'Success.';
8  }
9  else {
10         echo 'Insertion failed. Please try again.';
11 }
12 ?>
13
```

Kuva 4. Opiskelijan nimen syöttö. Riveillä kaksi ja kolme koodi ottaa yhteyttä MySQL-tietokantaan ja valitsee oikean koulun, jonka sisään koodi syöttää opiskelijan nimen rivillä viisi.

PHP on yleensä käyttäjältä näkymättömissä. Apache lähettää dynaamiset PHP -sivut PHP:n prosessoitavaksi, jolloin tulokseksi saadaan staattinen sivusto, jota käyttäjä pääsee katsomaan. Joskus kuitenkin joko Apachessa tai PHP:ssä on ongelmia, jolloin sivun PHP-koodi tulee näkyville. Tästä täysin avoimesta koodista on hyvin helppo etsiä tietoturva-aukkoja aivan alkeellisimmillakin tietotaidoilla. Lisäksi koodin sisällä saattaa olla linkattuna konfigurointitiedostoja, joiden linkkejä seuraamalla käyttäjä pääsee käsiksi sivuston juureen, jota kautta hän voi jatkaa matkaansa minne sivuston tiedostopuun haarakkeisiin tahansa. Koska PHP:n tai Apachen toimintaan ei voida vaikuttaa, voidaan ainoastaan järjestellä tiedostopuu siten, ettei käyttäjä vahingossakaan pääse käsiksi mihinkään kiellettyyn dataan. Kaikkien tiedostojuuressa sijaitsevien tiedostojen siis tulisi kestää päivänvaloa. (PHP Freak 2008.)

### 2.3.4 JavaScript-tietoturvariskit

JavaScript on monikäyttöinen, laajalle levinnyt ja tehokas koodikieli, jonka avulla voidaan helpottaa käyttäjän elämää esimerkiksi varoittamalla väärin menneistä salasanoista tai auttamalla käyttäjää täyttämään kaavakkeita. JavaScript on kuitenkin vaarallinen työkalu väärissä käsissä. Sillä voidaan kerätä selaimen evästeitä ja lähettää niitä eteenpäin, ja JavaScriptillä on aina täydet oikeudet lähteestä riippumatta, jolloin vihamielisen koodin lisäämistä ei voida estää.

JavaScript on helposti kaikkien käyttäjien nähtävillä, aivan kuten html. Tämän takia JavaScriptiä ei koskaan tulisi käyttää valonaran tiedon, esimerkiksi luottokorttitietojen tai evästeiden säilömiseen tai tiedon suojaamiseen ja validointiin. Hakkerit ja krakkerit saavat JavaScript-suojauksen helposti laitettua pois päältä, jolloin tieto jää täysin suojaamatta ja käyttäjät voivat lisätä kommentti- tai kaavakekenttiin mitä tahansa. Ikinä ei pitäisi myöskään luottaa oman serverin ulkopuoliseen JavaScriptiin, sillä sen sisällön turvallisuudesta tai pysyvyydestä ei ole mitään takeita.

## 2.4 Tietoturva käyttäjän kannalta

Tietokoneet ja Internet ovat arkipäiväistyneet tavallisten ihmisten elämässä jo siihen pisteeseen, että vain pieni osa suomalaisista ei käytä Internetiä lähes päivittäin. Laajakaistayhteys löytyy nykyään reilulla 70 prosentilla suomalaisista kotitalouksista (Tilastokeskus 2012), ja Internetiä käyttää päivittäin 16–44-vuotiaista 92 prosenttia ja 45–74-vuotiaistakin vielä 60 prosenttia, joskin 65–74-vuotiaiden käyttöprosentti jääkin noin kolmeenkymmeneen (Yle 2011). Tämän seurauksena myös tietoturvavalistus on edistynyt huimasti, ja jo ala-asteikäisille opetetaan turvallisen ja vastuullisen Internetin käytön perusteita. Esimerkiksi Tietoturvakoulu tarjoaa vanhemmille ja opettajille informaatiota ja välineitä mediakasvatusta varten, ja järjestää vuosittain helmikuussa Tietoturvapäivän, jonka tarkoituksena on valistaa tavallisia ihmisiä sähköisten medioiden turvallisesta käytöstä (Tietoturvakoulu 2013). On siis melko turvallista olettaa, että suurimmalla osalla ihmisistä on jonkinlainen käsitys tietoturvan perusteista, mutta on silti vaarallista jättää käyttäjän tietoturvasta huolehtimista kokonaan käyttäjän itsensä hartioille. On myös monia tapoja, joilla voidaan luoda asiakkaalle turvallinen ympäristö tarjoamalla kaikki tarpeellinen informaatio.

Ensimmäinen käyttäjälähtöinen tietoturvan kompastuskivi ovat salasanat. Ihmisillä on jo monta eri salasanaa ja numeroyhdistelmää muistettavana tietokoneen salasanoista

aina pankkikortin pin-koodiin, joten ihmiset turvautuvat usein käyttämään uutena salasanana jotain heille tuttua sanaa, jota eivät usko muiden arvaavan. Tämän lisäksi ihmiset käyttävät samaa salasanaa useilla eri sivustoilla ja joskus joka saman käyttäjätunnuksen parina. Valitettavan usein sana on jo itsessään helposti arvattava, jotta se olisi käyttäjän helpompi muistaa, ja vaikka sana olisikin täysin mielivaltaisesti valittu, se voidaan helposti murtaa käyttäen brute force -tekniikkaa. Jos salasanan rakentamiseen on käytetty pelkkiä kirjaimia ja salasanan pituus on esimerkiksi kahdeksan merkkiä, saa brute force salasanan selville muutamissa sekunneissa. Salasanan varmuutta ei juurikaan paranna, vaikka osan kirjaimista korvaisi numeroilla tai numeroita lisäisi salasanan perään, sillä tämä ei ole millään lailla uusi tai omaperäinen idea ja on jo monien hakkereiden tiedossa. Tämän takia käyttäjää tulisi kannustaa luomaan mahdollisimman pitkä salasana, jonka pituutta ei ole ennalta rajattu, ja joka koostuu useammasta käyttäjälle itselle järkevistä sanoista. Esimerkiksi Tr0ub4dor&3-salasanan muistaminen on vaikeaa, mutta koneelle arvaamiseen kuluu aikaa noin kolme päivää, jos arvausnopeus on 1000 arvausta per sekunti, kun taas ”correct horse battery staple” -yhdistelmän murtamiseen kuluisi samalla nopeudella 550 vuotta, mutta on huomattavasti helpompi käyttäjälle muistaa, esimerkiksi vinkkikuvan piirtämisen avulla (Munroe 2012).

Toinen salasanaan liittyvä ongelma on salasanan tallentaminen evästeisiin. Useilla sivustoilla voidaan klikata ”muista minut” -nappia, jolloin salasana sekä käyttäjätunnus tallentuvat sivustolle. Tämä toki nopeuttaa sivustolla käymistä, kun ei jokaisella kerralla tarvitse erikseen kirjautua sisään, mutta tällöin tiedot jäävät myös krakkereiden saataville, varsinkin jos salasanojen talletuksessa ei ole käytetty minkään tasoista kryptausta, jolloin vain salausavaimen tunteva taho saa sen tietoonsa. Näiden riskien takia ei käyttäjille tulisi edes tarjota mahdollisuutta pysyä kirjautuneena, sillä vaikka käyttäjä olisikin tietoinen riskeistä, ihmiset usein uskottelevat itselleen, ettei heille voi tapahtua mitään pahaa.

# 3 SIVUSTON SUUNNITTELU JA RAKENTAMINEN

## 3.1 Functional Safety Finland Oy:n vanhat sivut

Yrityksellä oli projektin alkaessa jo olemassa olevat sivut, jotka yrityksen perustaja Heikkilä oli tehnyt. Heikkilä oli syvästi pettynyt sivuston huonoon käytettävyyteen, vanhanaikaiseen ulkoasuun ja sisällön puutteeseen. (Liite 1.)

Vanhan sivuston värimaailma on tunkkainen ja mitäänsanomaton. Sisällön taustavärinä käytetty vaalea turkoosi on sinällään hyvä väri, mutta sen yhdistäminen tumman harmaaseen taustaan ja kirkkaan keltaiseen navigaation korostusväriin ei luo yrityksestä osaavaa ja luotettavaa kuvaa. Lisäksi leipätekstinä käytetty kirjoituskoneen kirjasinta muistuttava Consolas-fontti saa sivuston näyttämään keskeneräiseltä ja koruttomalta. Koska navigaatiossa on käytetty samaa fonttia vain hieman suuremmalla pistekoolla, se ei erotu muusta sisällöstä tarpeeksi, vaikka navigaatiossa onkin käytetty suuraakkosia eli versaalia (Typografia: Kieltä vai visuaalisuutta? 2001, 156) ja aktiivisen linkin taustalla on keltainen kuvio.

Sivuston logo on sijoitettu yleisestä normista poiketen sivun oikeaan yläkulmaan vasemman yläkulman sijaan (Krug 2006, 64), jolloin logo jää helposti käyttäjältä huomaamatta. Etusivulle vievä linkki on nimetty Functional Safety Finlandiksi, jolloin firman nimi on sivuston oikeassa yläkulmassa, jolloin käyttäjät tietävät, millä sivustolla ovat. Tästä kuitenkin valitettavasti seuraa se, että ihmiset, jotka etsivät pääsyä takaisin etusivulle, eivät sitä välttämättä löydä epäkäytännöllisen linkin nimen takia.

Navigaatio on selkeä ja yksinkertainen, osin sivuston suppeuden takia. Aktiivinen sivu on merkitty selkeästi keltaisella taustavärillä, ja kun kursori asetetaan linkin ylle muuttuu linkki valkoiseksi. Tämä auttaa käyttäjää ymmärtämään, että tekstiä voi klikata. Linkit on nimetty selkeästi ja yksiselitteisesti, ja linkin nimi vastaa siitä aukeavan sivun otsikkoa, jolloin käyttäjä tietää klikanneensa oikeaa kohtaa (Krug 2006, 73). Logoon ei ole kuitenkaan sisällytetty etusivulle johtavaa linkkiä. Vaikka tämä ei ole välttämätöntä, huomasin itseni vahingossa klikkailevan logoa päästäkseni takaisin aloitussivulle. Huomasin myös, että selaimen välilehdessä näkyvä teksti koostuu vain aktiivisen sivun nimestä ja siitä puuttuu täysin firman ja näin myös sivuston nimi. Tämän takia hukkasin sivuston kymmenien muiden auki olevien toiminnallista turvallisuutta käsittelevien välilehtien joukkoon, ja sitä oli myös vaikea löytää sivuhistoriasta ja kirjanmerkeistä.

Sisällöltään ja toiminnoiltaan sivusto on hyvin rajoittunut. Etusivulla sijaitsevan koulutus- ja palvelutarjonnan lisäksi sivustolta löytyvät yhteystietosivu ja toiminnallisen turvallisuuden määritelmä. Yhteystietosivulta löytyy yhteystietojen lisäksi tietoa Heikkilän työhistoriasta ja -osaamisesta sekä kopio toiminnallisen turvallisuuden koulutus-

sertifikaatista. Valitettavasti yhteystiedot eli käynti- ja sähköpostiosoitteet sekä puhelinnumero on esitetty kehystetyin kuvin, jolloin sisältöä ei pysty kopioimaan leikepöydälle ja käyttäjän on pakko kirjoittaa tiedot ylös käsin. Kuville on muistettu määritellä alt-merkinnät, joista kuvien sisältö löytyy tekstimuodossa, jolloin informaatio ei jää näytönlukijaa hyväksikäyttävät tai kuvat selaimesta estäneiltä käyttäjiltä saamatta (Krug 2006, 179). Valitettavasti sertifikaatti ei ole saanut yhtä selkeää alt-määritelmää. Määritelmässä lukee vain ”certificate”, jolloin sertifikaatin idea ja tarkoitus saattavat jäädä käyttäjiltä kokonaan hämäräksi. Tietenkään sertifikaatin koko sisältöä ei kannata alt-määreeksi laittaa, mutta määrettä olisi voinut tarkentaa mainitsemalla kyseessä olevan toiminnallisen turvallisuuden sertifikaatti ja kirjaamalla ylös sertifikaatin luovuttanut taho ja saavuttamisvuosi.

Yhteystietosivulla oleva kuva Heikkilästä ei myöskään ole kovin edustava. Kuva on huonosti syvätty eli irrotettu taustastaan, jolloin jäljelle jää terävä, luonnoton kontrasti kuvan ja turkoosin taustavärin välillä. Toiminnallisen turvallisuuden määrittelevä sivu ei valitettavasti onnistu avaamaan termiä alaa täysin tuntemattomalle lukijalle. Tosin muistakaan lähteistä ei termille kovin selkeää ja yleismaailmallista määritelmää löytynyt, joten syy ei välttämättä ole sivuston tekijän.

## 3.2 Sisällön suunnittelu

Heti projektin alussa Functional Safety Finlandilla oli tarkka näkemys siitä, mitä sisältöä sivuillensa toivoo. Ensisijaisina sisältöinä olivat yrityksen esittely- ja yhteystiedot, koulutuspalvelut, referenssiasiakkaiden esittelyt ja yrityksen toimintaan liittyvät ajankohtaiset uutiset. Näillä sisällöillä saataisiin jo aikaan tarpeeksi informatiivinen sivusto, jotta siitä olisi yrityksen toiminnalle hyötyä. Ehtona tietenkin on, että informaatio on ammattimaisesti ja selkeästi tuotettua. Epäselvästi jäsennelty, kirjoitusvirheitä vilisevä teksti antaa kirjoittajasta, yrityksestä ja sivuston ylläpitäjästä vähä-älyisen ja kouluttamattoman kuvan. Tätä tulisi tietysti kaikkien sisällöntuottajien välttää, mutta erityisesti nyt, kun yrityksen toiminta perustuu turvallisuuteen ja sen tuomaan luottamukseen. Lisäksi sivuston tekstisisältöön huonosti tai ei lainkaan sopivat kuvat myös osaltaan vaikeuttavat käyttäjän sivustoon sitouttamista, sillä jos käyttäjä joutuu edes hetkeksi miettimään kuvan ja tekstin mahdollista yhteyttä, käyttäjän huomio kiinnittyy yrityksen ja sivun viestin kannalta epäolennaisiin asioihin.

Sivuston sisältö on hyvin pitkälle tekstipohjaista suuren informaatiomäärän ja puutteellisen aikaisemman materiaalin takia. Yrityksellä ei aiemmin ole ollut tarvetta hyvin tiiviiseen tekstiin esimerkiksi esitteiden tai mainosten suunnittelua varten. Tämä on haitaksi verkkosivuston sisältöä tuotettaessa, sillä verkossa on vaikea saada käyttäjä lukemaan pitkiä tekstejä. Käyttäjät ovat taipuvaisia vain selaamaan otsikot ja kenties kappaleiden ensimmäiset sanat etsiessään itseään kiinnostavaa informaatiota. (Krug 2006, 46.) Käyttäjät ovat tottuneet tehokkaaseen informaation kulkuun, nopeasti latautuviin sivuihin ja jatkuvaan kiireeseen, joten sisällön on oltava koskettavaa tai hyödyllistä, jotta käyttäjä saataisiin pysähtymään ja keskittymään sisältöön. (Mts, 22.)

Sain Functional Safety Finlandilta yrityksen esitteisiin tarkoitetut tekstit, joista voisin muokata sivuston sisällön. Esimerkiksi tarjolla olevista palveluista oli runsaasti tekstiä, mutta onneksi siitä oli helppo poimia ingressejä, jotka aiheeseen sopivan kuvan kanssa toimivat tehokkaana sisäänheittojuttuna yleisellä palvelut-sivulla, jonka kautta pääsisi lukemaan palveluiden täyspitkät kuvaukset. Tästä huolimatta tekstiä oli aivan liian paljon, mutta aihepiirin ja palveluiden monimutkaisuuden takia niistä oli vaikea karsia paljoakaan. Päädyin siis tekemään ingresseistä mahdollisimman itsestään selviä ja kaiken kattavia, jotta lukija saisi kaiken pakollisen tiedon avaamatta yksittäisen palvelun omaa sivua.

## 3.3 Sisällönhallintajärjestelmän ja webhotellin valinta

### 3.3.1 Sisällönhallintajärjestelmä

Sisällönhallintajärjestelmä tai julkaisujärjestelmä (eng. content managment system, CMS) on järjestelmä, jolla kyetään hallinnoimaan verkkosivuston sisältöä ja dokumentteja ilman teknistä tai koodikielten osaamista (Tolvanen 2009). Ensimmäinen sisällönhallintajärjestelmä julkaistiin 1990-luvun lopulla, ja sillä pyrittiin helpottamaan moninaisten eri koodiversioiden kirjoittamista. Sisällönhallintajärjestelmät ovat hyödyllisiä esimerkiksi erikokoisten organisaatioiden verkkosivujen pohjana, sillä silloin kuka tahansa organisaation sisällä pystyy muokkaamaan ja lisäämään verkkosivujen sisältöä ilman IT-osaston osallistumista. Tiedostojen siirto palvelimelle onnistuu vaivattomasti ja melkein kaikkien sisällönhallintajärjestelmien sisältämällä WYSIWYG-tyyppisellä (eng. what you see is what you get eli mitä näet, sitä saat) tekstinkäsittely-

käyttöliittymä on mahdollisimman yksinkertaista. WYSIWYG-käsittelyohjelmissa pyritään näyttämään muokattava dokumentti juuri sellaisena, miltä se lopullisessa muodossaan näyttää. Ongelmaksi saattaa kuitenkin verkkojulkaisuissa muodostua se, että esimerkiksi kaikki kirjasintyypit eivät näy selaimessa.

Nykyään sisällönhallintajärjestelmiä on useilta eri valmistajilta, ja jokainen niistä soveltuu hieman eri käyttötarkoitukseen. Kaikista laajimmin käytössä oleva WordPress soveltuu parhaiten blogien ylläpitoon, kun taas parhaan avoimen lähdekoodin sisällönhallintajärjestelmän palkinnon voittanut Joomla (Packt Publishing 2011) on suunniteltu suurten, monimutkaisten sivustojen pyörittämiseen. Tästä huolimatta sisällönhallintajärjestelmien välillä vallitsee kova kilpailu käyttäjien huomiosta, jolloin jokainen vielä toiminnassa oleva sisällönhallintajärjestelmä pyrkii pysymään jatkuvasti teknologian aallonharjalla ja mahdollisimman helppokäyttöisenä ja turvallisena. Kuitenkin on aina muistettava se, ettei mikään järjestelmä ole täysin turvallinen. Suuret, useita miljoonia käyttäjiä palvelevat hallintajärjestelmät ovat tietojärjestelmiin murtautuvien krakkereiden lempikohteita. Esimerkiksi WordPress pyörittää yli 57 miljoonaa sivustoa tällä hetkellä (WordPress 2012). Lohdullista on tosin se, että valtaosa sisällönhallintajärjestelmien turvallisuusongelmista johtuvat huolimattomista käyttäjistä ja heikkorakenteisista, amatöörien rakentamista liitännäisistä. Esimerkiksi Wpmu.org (Kaiser 2012) kirjoitti National Vulnerability Databasen tilastoista, joiden pohjalta pystyttiin laskemaan, että noin 40 % WordPressin tietoturvaongelmista johtui suoraan kolmannen osapuolen tuottamista plugineista.

Lähtiessäni etsimään mahdollisimman täydellistä sisällönhallintajärjestelmää karsin ehdokkaat ensin sen perusteella, mitä käyttäjät niistä sanoivat. Keskustelin aiheesta opiskelijatovereideni sekä verkkoviestinnän alalla työskentelevien ystävieni kanssa ja pyysin heidän mielipiteitään. Lopullisiksi vaihtoehdoiksi jäivät WordPress, Joomla, Drupal ja SilverStripe. WordPress on monille jo tuttu ja helppokäyttöinen, Joomla ja Drupal monikäyttöisiä ja -muotoisia ja SilverStripe nopeasti kehittyvä musta hevonen.

WordPress putosi melko nopeasti vaihtoehdoista pois kaikista hyvistä puolistaan huolimatta, koska epäilin lähinnä blogien ja pienten sivustojen ylläpitoon suunnitellun järjestelmän soveltuvan huonosti verkkosivuston pyörittämiseen tilanteessa, jossa sivustoa laajennettaisiin tulevaisuudessa rajustikin. WordPress oli kuitenkin sinnitellyt lis-

talla kauan, sillä WordPress on suuren käyttäjäjoukkonsa takia äärimmäisen monipuolinen, kaikkiin mahdollisiin ongelmiin on helppoa löytää vastaus ja liitännäisiä löytyy jokaiseen tarpeeseen. (Quinn & Gardner-Madras 2010.)

Koska asiakkaanani toimiva Functional Safety Finland tarjoaa toiminnallisen turvallisuuden palveluita, on yrityksen imagon kannalta parasta, että kaikki, mikä uuteen sivustoon liittyy, on rakennettu mahdollisimman turvalliseksi. Tästä syystä esitin asiakkaalleni toiveen, että voisimme valita käyttöjärjestelmäksemme jonkin pienemmän kokoluokan järjestelmän, jota krakkerit eivät niin innokkaasti havittelisi. Asiakas oli alun perin itse käyttänyt Joomlaa, mutta sekin on kokoluokkansa ja hyvän maineensa takia krakkereiden kiikarissa. Lisäksi Joomla on minulle täysin vieras ja rakenteeltaan kilpakumppaneitaan monimutkaisempi, jolloin sen opettelu olisi vienyt turhaa aikaa projektista. Myös Drupal on saanut huonoa palautetta sekavasta käyttöliittymästään ja liian plugin-keskeisestä rakenteestaan. Drupalin runko on hyvin paljas, jolloin yllättävän suuri osa sivuston toiminnallisuudesta joudutaan toteuttamaan pluginien kautta, toisin kuin esimerkiksi WordPressissä. Päädyimme lopulta käyttämään SilverStripeä juuri sen vaatimattoman koon sekä nopean kehityksen takia. Noin 0,07 % sivustoista käyttää SilverStripeä alustanaan, verrattuna Wordpressin 64,96 %:n (Built With 2013a). SilverStripe on kasvattanut osuuttaan vuoden takaisesta 0,02 prosenttiyksikköä (Built With 2013b.)

### 3.3.2 Webhotellit

Webhotelli on ”*yritys, joka myy omilta palvelimiltaan tilaa asiakkaiden www-sivuille*” (Järvinen 2004, 234). Toisin sanoen sekä yritykset että yksityiset henkilöt voivat ostaa verkkosivuilleen tarvittavan ylläpitotilan ulkopuoliselta yritykseltä, joka huolehtii palvelinten ylläpidosta ja tietoturvasta asiakkaan puolesta. Nimensä mukaisesti palvelun hinta nousee sitä mukaan, mitä kauemmin verkkosivustoa serverillä pidetään, webhotellit laskuttavat asiakkaitaan joko kuukausi- tai vuosierissä. Viime vuosien aikana webhotellien tarjonta on kasvanut huimasti kysynnän mukana niin Suomessa kuin ulkomaillakin. Tämän sekä teknologian nopean kehityksen takia hinnat ovat myös laskeneet nopealla tahdilla, joten melkein kuka tahansa voi turvautua webhotellien palveluihin.

Webhotellien suuren tarjonnan takia on hieman vaikeaa etsiä sopivinta. Onneksi verkosta löytyy useita webhotellilistoja, joiden sisältö perustuu käyttäjiltä saatuihin arvioihin. Kävin aluksi läpi itselleni tuttuja webhotelleja, mutta tyytymättömänä tarjontaan etsin erään tällaisen listan käsiini. Aluksi löysin erittäin kattavan listan ulkomaalaisista webhotelleista, jotka olivat keskimäärin suomalaisia halvempia tarjoten kuitenkin samat palvelut. Keräsin luotettavimman ja sopivimman oloiset webhotellit palveluineen taulukkoon ja lähetin Functional Safety Finlandille. He olivat kuitenkin sitä mieltä, että palveluntarjoajan tulisi olla suomalainen. Tätä he perustelivat mm. sujuvammalla asiakaspalvelulla, luotettavammalla ylläpidolla sekä suomalaisten mittapuulla paremmin ajoitetuilla huoltokatkoilla, sillä serverien huoltokatkot ajoitetaan usein yöaikaan. Lähdin siitä keräämään uutta listaa käyden läpi Webhotellivertailuun (2013) kirjoitettuja kommentteja ja arvosteluita. Käyttäjien mielestä luotettavin ja miellyttävin webhotelli on Hostingpalvelu.fi, vaikka ääniä ei montaa vielä olekaan. Päätimme lopulta siirtyä Radicenterin asiakkaiksi. Syynä päätökseen oli riittävän suuri 5 Gt:n levytila, rajoittamaton liikenne- sekä tietokantamäärä sekä päivittäinen varmuuskopiointi. Lisäksi kaupan päälle tuli mukaan jatkuva SSH- eli Secure Shell -yhteys, joka on tietoliikenteen salaamiseen tarkoitettu protokolla. Käytännössä se suojaa ftp- ja http-liikennettä serverin ja käyttäjän välillä. Radicenter myös tarjoaa mahdollisuuden ostaa SSL-sertifikaatin (ks. luku 4.2).

## 3.4 Verkkosivuston rakenne ja hierarkia

Asiakkaan toiveena oli luoda helppokäyttöinen ja luokseen kutsuva sivusto, jonka käytettävyyteen panostettaisiin erityisen huolellisesti. Tämä olisi tietysti hyvä tavoite minkä tahansa sivuston rakentamisessa, mutta erityisesti toiminnallisen turvallisuuden yritykselle selkeän käytettävyyden luoma turvallisuuden tunne on hyvää mainosta ja parantaa yrityksen imagoa. Selkeä käytettävyys saadaan suunnittelemalla sekä sivuston rakenne että hierarkia harkiten ja huolella vallitsevien konventioiden mukaisesti.

Konventiot ovat vakiintuneita käytäntöjä, jotka syntyvät, kun tarpeeksi monet verkkosivustot on rakennettu vakiintuneille paikoille asetetuista osista. Näin voidaan luottaa siihen, että ennenkin Internetiä käyttäneet ihmiset löytävät varmemmin etsimänsä ja verkon ensikertalaiset oppivat verkon lainalaisuudet. Konventioiden lisäksi selkeä hierarkia on käytettävyyden kulmakivi. Hierarkian saa selkeäksi korostamalla sivun ymmärtämisen kannalta tärkeitä asioita, kuten otsikoita, ympäristöään suuremmalla

koolla, ympäröivällä tyhjällä tilalla ja väreillä. Loogisesti toisiinsa kuuluvat objektit, kuten navigaation linkit, yhdistetään toisiinsa visuaalisen asettelun ja ulkomuodon avulla, jolloin ne ovat helpommin havaittavissa. Lisäksi sivuston osiin saadaan selkeyttä asettelemalla tärkeät objektit sivun ylälaitaan ja vasemmalle, vähemmän tärkeät alas. Sivun ylälaidan objektit huomataan ensin ja mahtuvat suuremmalla todennäköisyydellä näytölle ilman vieritystä. Mitä selkeämmin hierarkia saadaan toteutettua, sitä vähemmän käyttäjän tarvitsee miettiä tekemisiään. (Krug 2006, 31–36.)

Saadakseni pidettyä sivuston mahdollisimman selkeänä ja yksinkertaisena päätin pitäytyä hyväksi havaituissa konventioissa. Näin pystyisin varmistamaan, etteivät sivuston ulkoasu ja rakenne veisi huomiota sisällöltä, joka on tämän projektin kaltaisissa informatiivisissa sivustoissa ensisijaisen tärkeää. Sijoitin yrityksen logon sekä liiketunnuksen vasempaan yläkulmaan ja navigaation vaakatasoon logon alapuolelle. Aluksi sijoitin hakupalkin navigaation rinnalle, mutta navigaation täyttyessä päätin siirtää hakupalkin oikeaan yläkulmaan. Muutos ei vaikuta käytettävyyteen merkittävästi, sillä vaikka yleinen käytäntö onkin asettaa hakupalkki oikeaan ylälaitaan, navigaation oikea reuna olisi ollut tarpeeksi korkealla, jotta käyttäjä löytäisi sen ilman ylimääräistä silmäilyä. Eräässä rautalankamallissa varasin navigaation alapuolelle tilaa mahdollisille alalinkeille, jos sivuston sisältö niin vaatisi.

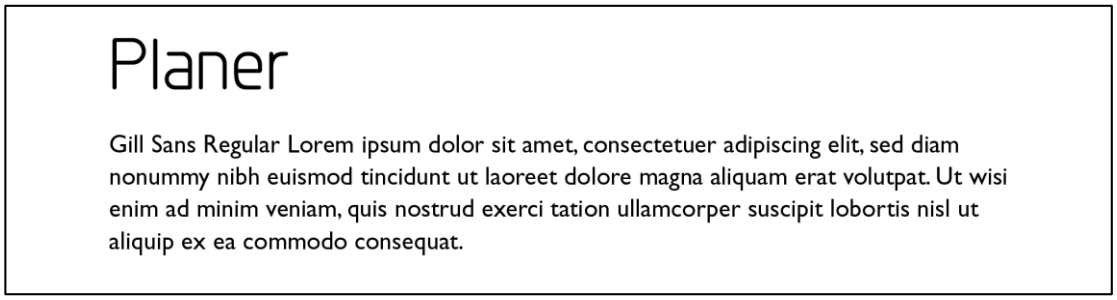
Kaikkien tekemieni rautalankamallien etusivua dominoi suuri kuvaelementti, johon oli liitetty tekstiä. Kahdessa ensimmäisessä mallissa kuvaelementtinä toimi diaesitys, jossa kuvat teksteineen vaihtuisivat muutaman sekunnin välein. (Liite 2.)

## 3.5 Ulkoasu, värimaailma ja typografia

Melkein heti projektin käynnistyttyä sain Functional Safety Finlandin graafiseen ohjeistoon. Graafisen ohjeiston oli suunnitellut opinnäytetyönään Kymenlaakson ammattikorkeakoulun graafikko-opiskelija Sofia Herkiä. Materiaali ei valitettavasti ollut vielä täysin viimeistelty, joten jouduin valmistautumaan mahdollisiin pieniin ulkoasumuutoksiin. Osa painomateriaaleista oli vielä suunnittelematta tai vasta ajatusasteella. Mitään mullistavia muutoksia ei kuitenkaan ollut odotettavissa, Functional Safety Finlandilla oli selkeä ja vahva visio unelmiensa verkkosivujen ulkoasusta.

Graafisen ohjeiston mukaan käytettävissäni olivat puhdas musta, vaalea harmaa ja puhdas keltainen. Lisäksi ulkoasussa pystyy käyttämään harmaan eri valööriasteita,

tosin vain vaaleanharmaasta tummempaan päin. Värit olivat aihealueeseen sopivat ja loogiset sekä huomiota herättävät. Ensimmäinen huolenaiheeni oli se, miten saisin värit siirrettyä verkkoon tekemättä sivustosta liian tummaa tai aggressiivista. Verkkojulkaisujen otsikkona toimii Planer ja leipätekstin fonttina Gill Sans Regularia. (Kuva 5.)




Kuva 5. Planer ja Gill Sans Regular.

Ensimmäinen luonnokseni oli hieman graafisen ohjeiston ulkoasua jäljittelevä, ja koostui kolmesta mustalla taustalla olevasta valkoisesta, pyöristetystä neliöstä, joiden reunoja koristivat varjostukset. Ylin neliö oli omistettu navigaatiolle ja logolle, keskimmäinen neliö sisällölle ja viimeinen alatunnisteelle, johon kuuluisi yrityksen yhteystiedot. Sisällön kehys oli jaettu kolmeen osaan, joista suurimman osan vei diaesitys, jossa esiteltäisiin yrityksen arvoja ja visioita. Diaesityksen oikealla puolella sijaitsi koulutuskalenteri, joka näyttäisi kuluvan ja seuraavan kuun koulutustilaisuudet. Koulutuspäivää klikattaessa avautuisi pieni, muun sisällön päällä leijuva Lightbox-ikkuna, josta löytyvät koulutustilaisuuden tarkemmat tiedot. Diaesityksen ja koulutuskalenterin alapuolelle sijoitin pieniä laatikoita, joihin pystyisi sijoittamaan pieniä tietoiskuja, kuten suosituimpia koulutuspaketteja. Navigaation fonttina käytin Planeria. Pääikkunan jaottelu oli lopullista viilausta vaille onnistunut, mutta ulkoasu oli tunkkainen ja vanhentunut. Laatikoiden varjostukset eivät sopineet lainkaan Functional Safety Finlandin tavoittelemaan moderniin ilmeeseen. (Liite 3.)

Toinen version pääikkuna oli samanlainen kuin edellisessä versiossa, mutta navigaatio ja alatunniste sijaitsivat koko sivun levyisistä mustista laatikoista, joissa oli kevyet varjostukset. Sivun taustavärinä oli valkoinen, ja pääikkuna oli kehystetty varjostuksilla, jotta se erottuisi taustastaan. Koko sivun levyinen navigaatiopalkki antoi linkeille ja logolle enemmän tilaa hengittää ja sai sivun näyttämään vakaammalta. Koska navigaation pohjavärinä oli musta, piti tekstin väri vaihtaa valkoiseksi. Tämä kuitenkin vaikeutti navigaation lukemista, sillä käytetty Planer-fontti oli valitettavan ohut, eikä negatiivisena ollut tarpeeksi selkeä. Lisäksi kun korvasin liikemerkin ja Functional

Safety Finland -logon yhdistetyllä logolla, logon ja oikeassa yläkulmassa sijaitsevan hakupalkin väliin jäi häiritsevä tyhjiö. (Liite 4.)

Kolmas ulkoasuversio oli variaatio edellisestä, jossa ylätunnisteen taustavärinä oli tumma harmaa ja päänavigaatio oli erotettu omaan pieneen laatikkoonsa ylätunnisteen ja pääikkunan välille. Navigaation taustana oli pyöristetty, valkoinen suorakaide, joka oli selkeyden vuoksi varjostettu. Alatunnistetta ei ole lainkaan. Pääikkunasta puuttuivat varjoreunukset, minkä vuoksi sisällölle jäi enemmän tilaa. Sivusto jäi hallitsevan harmaan värin takia todella mitäänsanomattomaksi, eikä siinä ollut kunnollista kiintopistettä. Logon keltainen ja valkoinen teksti hävisivät taustaansa. Tosin sivu näytti raikkaammalta ilman pääikkunan varjoreunuksia, joten reunat jäivät myöhemmistä versioista pois. (Liite 5.)

Ulkoasuversioista suosikiksi nousi toinen versio, joten ryhdyin jalostamaan ideaa pidemmälle. Asettelin diaesityksen teksteineen ja diavalikoineen paikoilleen ja etsin siihen sekä tietoiskulaatikoihin kuvituskuvat. Sijoitin diaesityksen ja tietoiskulaatikoiden sisältötekstin läpikuultavalle valkoiselle pohjalle kuvan oikean puolen päälle. Lisäsin ylä- ja alatunnisteisiin sekä sivuston taustaan itse tekemäni harjatun alumiinin tekstuurin materiaalia elävöittämään. Kalenterin reunuksena olivat rinnakkain harmaa ja keltainen viiva, ja taustaväriksi valitsin harmaan. Kalenterin ruudukko oli tummemman harmaa, ja päivät, jolloin koulutuksia järjestetään, merkattiin keltaisella. Alatunnisteeseen laitoin yrityksen käyntiosoitteen ja puhelinnumeron. Paksunsin myös hieman navigaation fonttia lisäämällä siihen ohuen valkoisen reunuksen, sillä Planerista ei löytynyt lihavoitua versiota. (Liite 6.)

Työn edetessä ulkoasu muuttui vielä hieman. Vaalensin taustakuvaa hieman ja tummensin ylä- ja alatunnisteiden taustaa, jottei taustan tekstuuri olisi aivan niin selkeä. Lisäksi toisen tason navigaatio muuttui pudotusvalikosta vaakanavigaatioksi, samoin kuin käyttäjien kirjautumisikkuna, joka sijoitettiin aivan sivun yläreunaan. Poistin etusivulta myös kalenterin, mutta asiakas toivoi pystyvänsä lisäämään kalenterin tahtoessaan takaisin. (Liite 7.)

## 3.6 Sivuston toteutus

Kun ulkoasu oli päätetty, oli aika tutustua SilverStripeen. Minulle oli alustavasti selitetty, miten SilverStripen hierarkia toimii ja mitä kaikkea tulisi ottaa huomioon sivustoa rakennettaessa. Jos esimerkiksi tehdään muutoksia sivupohjiin linkitettyihin tiedostoihin, tulee sen näyttävän sivuston osoitteen perään kirjoittaa ?flush=1, jolloin SilverStripe lataa kaikki linkitetyt tiedostot uudestaan. Esimerkkinä, jos muokkaan etusivun diaesitystä, pitää osoiteriville kirjoittaa fsf.fi/sandbox/etusivu/?flush=1. Muuten muutokset eivät tule näkyviin.

Alkuvaikeuksien jälkeen sivuston rakentaminen alkoi sujua. Opin SilverStripen omat haku- ja if-lausekkeet nopeasti ja selkeät, aloittelijaystävälliset ohjeet auttoivat ensimmäisten sivupohjien luomisessa. Perusohjeiden lisäksi ei ongelmiin kuitenkaan löytynyt paljoa apua. Esimerkiksi moduulien asentamiseen ei löytynyt minkäänlaista tarkempaa ohjetta. Vaikka moduulien asentaminen, eli vain sivuston juureen purkaminen, oli yksinkertainen prosessi, ei mahdollisiin ongelmatilanteisiin löytynyt ohjeita. Lisäksi omatoiminen sivupohjien rakentaminen päättyi joka kerta sivuston kaatumiseen syystä, jota en vieläkään ymmärrä. Sivupohjien luomisen hankaluus oli suurin syy siihen, miksi sivuston valmistuminen viivästyi niin paljon.

Toinen suuri ongelma oli sisällönhallintapaneelin spontaani kaatuminen. Kaatumisen aiheutti kahteen kertaan jonkun sivun sivutyypin muuttaminen. Ensimmäisellä kerralla kaatuminen tapahtui, kun muutin sivutyypin itse rakentamakseni, uudeksi tyypiksi, ja oletin koodissani olevan jotain vikaa. Toisella kerralla kuitenkin kaatumisen aiheutti sivutyypin muuttaminen SilversStripen omaksi perussivuksi. Sivutyypin muuttaminen takaisin alkuperäisekseen kuitenkin korjasi aina ongelman. Kerran kuitenkin onnistuin täysin viattomasti kaatamaan koko sisällönhallintajärjestelmän, jonka korjaamiseksi ei auttanut muu kuin uudelleenasennus ja tietokannan palauttaminen. Onneksi olin kuitenkin huomannut ottaa tietokannasta varmuuskopion muutamaa päivää aikaisemmin.

Mikään näistä ongelmista ei olisi ollut niin tuhoista, jos SilverStripeä varten olisi kerätty kunnollista dokumentaatiota. Vähäistä, jo olemassa olevaa dokumentaatiota harvensi lisäksi se, että vuoden 2012 kesäkuussa julkistettu 3.0-versio uudisti hallintajärjestelmää niin rankasti, etteivät vanhat ohjeet enää toimineet. Apua etsiessä oli oltava alituisesti valppaana, miten vanhaa ohjetta lukee. Heikon dokumentaation vuoksi sivuston yläreunan kirkkaan keltainen kirjautuneiden käyttäjien navigaatiopalkki jäi

kesken. Tarkoituksena oli sijoittaa sinne kirjautuneille käyttäjille henkilökohtaiset linkit ja näyttää kirjautumattomille vain kirjautumisikkuna ja mahdollisuus käyttäjätunnuksen luomiseen.

Sivuston ulkonäkö muuttui paikoin sivustoa rakennettaessa. Alun perin kaksipalstaiseksi tarkoitettu koulutuspalvelut-sivu muuttui yksipalstaiseksi pitkien tekstien ja suurempien kuvien takia. Lisäsin muutenkin sivuilla olevien kuvien määrää, sillä muutoin tekstipalstojen leveys venyi miltei lukukelvottomaksi. Kuvattomaksi tarkoitettu laiteturva-sivu, jonka sisältö tulee pääasiassa olemaan pdf-oppaita, muuttui miltei tekstittömäksi galleriasivuksi. Se, olivatko muutokset parempaan, pitää testata käytettävyystestin avulla.

# 4 KÄYTETTÄVYYSTESTAUS

## 4.1 Käytettävyystestauksen perusteet

Pienimuotoinen käytettävyystestaus on yksinkertaista toteuttaa. Koska testauksen ei tarvitse seurata tieteellisen tutkimuksen sääntöjä, kuten otannan satunnaisuutta tai testien keskinäistä vertailukelpoisuutta, voidaan testiä muuttaa lennosta, jos jokin kohta siitä ei toimi. Jos esimerkiksi testin yhdessä osassa yritetään tutkia käyttäjätilin luomisen helppoutta, mutta käyttäjätilin luominen ei toimi lainkaan, voidaan seuraavien testaajien kohdalla hypätä luomisosio kokonaan yli. Kun käyttäjätilin luominen on saatu toimimaan, voidaan sitä koskeva testiosio suorittaa irrallaan. (Krug 2010, 13–14.)

Pienimuotoisimmastakin käytettävyystestauksesta on hyötyä varsinkin ennen sivuston julkistamista, sillä jokaisella sivulla on käytettävyysongelmia. Vaikka sivusto olisi rakennettu huolella ja ammattitaidolla, on suunnittelija kuitenkin vain yksi ihminen. Se, mikä suunnittelijan mielestä on hyvä ratkaisu, ei välttämättä sovi sivuston kohderyhmälle. Tämä on ongelma varsinkin sivustoissa, jotka ovat laajenneet vuosien mittaan moninkertaisiksi ilman välitestauksia. Onneksi kaikkein räikeimmät käytettävyysongelmat ovat helposti havaittavissa ja usein myös korjattavissa. Niiden löytämiseksi kuitenkin tarvitaan käytettävyystestausta, sillä suunnittelijat ja suunnittelijan työtä seuranneet osapuolet sokeutuvat työlle nopeasti. Käytettävyystestauksessa on myös se etu, että silloin suunnittelija pääsee seuraamaan, miten käyttäjät oikeasti käyttävät sivuja. Suunnittelijoilla on aina jonkin verran värittynyt kuva käyttäjistä, joko parempaan tai huonompaan suuntaan, ja mielikuva on usein hyvin kategorisoitunut. On hy-

vin vaikeaa kuvitella käyttäjiä yksilöinä tai edes pieninä ryhminä, usein käyttäjät muotoutuvat homogeeniseksi massaksi. Kolme testikäyttäjää riittää pieneen testaukseen, sillä suurella todennäköisyydellä jo kolme eri käyttäjää löytävät pahimmat käytettävyysvirheet. Parhaat tulokset saadaan toistamalla koe kuukausittain, mutta keskitymme nyt vain yhteen testaukseen. (Krug 2010, 39–44.)

Käytettävyystestausta on miltei mahdotonta suorittaa liian aikaisin. Usein ajatellaan, että sivuston pitää olla toimiva, että sitä voisi testata, mutta testata voi tehdä muutenkin. Voidaan esimerkiksi rakentaa interaktiivisia rautalankamalleja tai paperiprototyyppejä käyttäjille kokeiltavaksi, tai vain tulostaa etusivun ulkoasu paperille ilman sisältöä ja kysyä ihmisiltä, mitä he uskovat sivuston käsittelevän. Jos suunnittelija ei halua keskeneräistä työtään muiden kritisoitavaksi, voi testauksen kohteeksi ottaa samankaltaisia sivustoja. Tämän testin perusteella voidaan välttää muiden sivujen sudenkuoppia, mutta samalla voidaan vahingossa luoda uusia.

## 4.2 Testauksen sisällön suunnittelu

Ennen testauksen aloittamista minun on listattava muutamia tärkeimpiä tehtäviä, mitä käyttäjät tulevat suurimmalla todennäköisyydellä sivulla tekemään. Tehtävien kuvaukset ovat hyvin lyhyitä ja itsestään selviä, kuten vaikkapa "hae tietoa koulutuspalveluista". Seuraavaksi testaajalle keksitään tausta ja syy etsiä kyseistä tietoa, esimerkiksi "Olet keskisuuren yrityksen uusi johtaja, ja pyrkimyksenäsi on saattaa yrityksesi turvallisuusasiat ajan tasalle. Ensimmäiseksi asiaksesi otat henkilöstösi kouluttamisen". Tehtävät on hyvä lukea ulkopuoliselle, jotta varmistutaan niiden selkeydestä ja johdattelemattomuudesta. Tehtävät tulostetaan pienille paperilapuille.

Päätin ensin suorittaa vain yksinkertaisen ulkoasukyselyn, jossa näytän ihmisille sivuston lopullista ulkoasua ja kysyn, mitä he siinä näkevät ja mitä tunteita se heissä herättää. Tällainen minitesti on helppoa ja nopeaa toteuttaa, ja sen voi suorittaa vaikkapa palaverin päätteeksi tai kesken lounastauon.

Päätin jo aikaisin ottaa testikäyttäjikseni pääosin muita Kymenlaakson ammattikorkeakoulun opiskelijoita. Koska teen käytettävyystestauksen koulutuntien aikana, ei minun tarvitse huolehtia testikäyttäjien keräämisestä, paikalle opastuksesta tai ajankäytön korvauksista. Lisäksi olin varma, ettei suurin osa opiskelijoista ole koskaan kuullutkaan toiminnallisesta turvallisuudesta, joten pystyin samalla tarkistamaan, että navigaation nimikkeet olivat selkeitä. Jos testikäyttäjät antavat luvan, nauhoitan hei-

dän kommenttinsa, jolloin minun ei tarvitse jakaa huomiotani testikäyttäjän ja muistiinpanojen kirjoittamisen välillä. Jos äänenlaatu on tarpeeksi hyvä, tallentuvat samalla testikäyttäjän tunteenilmaukset.

Testin aikana on tärkeää olla auttamatta testaajaa. On helppoa olla antamatta suoria ohjeita, mutta jo pelkät eleet, äänenpainot ja huolimattomat sanavalinnat ohjaavat testaajan käytöstä. On siis hyvä idea istua hieman testaajaa taaempana, mutta kuitenkin tarpeeksi lähellä, ettei mikään näytöllä näkyvä jää huomaamatta. Lisäksi on hyvä opetella pari fraasia, joilla vastata testaajan kysymyksiin ja huomioihin neutraalisti, kuten "mitä mietit", "tapahtuiko siitä sitä, mitä oletit" ja "miten toimisit, jos en olisi tässä". (Krug 2010, 23–89.)

## 4.3 Testauksen toteutus

Sivuston toteutuksen keskivaiheessa suoritin ensimmäisen käytettävyystestin, jossa näytin viidelle ihmiselle sivuston lopullisen ulkoasun ja pyysin heitä kuvailemaan sitä kolmella sanalla. Käyttäjät kuvailivat ulkoasua esimerkiksi sanoilla jämpti, tekninen ja selkeä, joten testi ei aiheuttanut ulkoasuun suuria muutoksi. Valitsemani kuvat saivat kuitenkin kritiikkiä osakseen, erityisesti diaesityksessä oleva kuva puhelimeen puhuvasta miehestä aiheutti hämmennystä yhdessä käyttäjistä, syystä jota hän ei osannut minulle artikuloida. Uskon hämmennyksen kuitenkin johtuvan eniten siitä, että diaesityksen tekstilaatikko peittää kuvassa olevan tietokoneen näytön, jolloin käyttäjä ei nähnyt, mitä mies sormellaan osoittaa. En olisi itse huomannut kiinnittää tällaisiin seikkoihin mitään huomiota, mutta testistä viisastuneena ymmärsin katsoa kuvia ja kuvien asettelua tarkemmin.

Huomasin testin jälkeen myös sen, etten osaa puhua neutraalisti. Monet lauseistani ja kysymyksistäni olivat johdattelevia, enkä niitä testitilanteessa huomannut. Kun esimerkiksi halusin tietää, miten suureksi käyttäjät kokevat sivuston takana olevan yhtiön, kysyin voisivatko nämä sivut kuulua suurelle yritykselle. Tällaiseen kysymykseen on helppo vain myötäillä ja nyökytellä, ja sainkin kysymykseeni positiivisen vastauksen. Nyt en kuitenkaan voi olla varma, tarkoittivatko käyttäjät sitä, mitä sanoivat. Parempi lausevalinta olisi ollut esimerkiksi ”Kuuluvatko sivut suurelle, keskisuurelle vai pienelle yritykselle?” tai ”Miten suureksi kuvittelisit tämän yhtiön?”.

Koska sivuston toteutukseen kului reilusti enemmän aikaa kuin olin kuvitellut, en ehtinyt suorittaa kunnollista käytettävyystestausta. Pystyisin periaatteessa testaamaan olemassa olevaa sivustoa, mutta koska sivusto ei nykyisellään ole kovin monimutkainen,

en kokenut testistä olevan paljoa hyötyä tässä vaiheessa. Kun saan rekisteröityneiden käyttäjien puolen toimimaan, on testi jo ajankohtaisempi, sillä rekisteröityneiden käyttäjien valikon asettelussa ja sisällössä sekä peruskäyttäjiltä piilotettujen sivujen sisällön kanssa on ollut ongelmia heti projektin alusta asti. Rekisteröityneiden käyttäjien toimintoja on pakko päästä testaamaan, ja toivoisin selviäväni yhdellä käytettävyystestillä ennen julkistamista, joten jätän tässä vaiheessa testin tekemättä. Voisin myös testata jonkun ulkopuolisen tahon verkkosivuja, mutta en ole vielä löytänyt tarpeeksi samankaltaista sivustoa. Tällainenkin testi voisi myös kuitenkin olla lähitulevaisuudessa paikallaan, jotta saadaan informaatiota sivuston mahdollisia laajennuksia varten.

# 5 YHTEENVETO

Functional Safety Finland -projekti oli kokonaisuudessaan antoisa ja opettavainen, joskin toisinaan turhauttava. Tietoturvaan tutustuminen avasi silmäni lukuisille Internetiä riivaaville riskeille, joiden tiedostaminen ja korjauksen yksinkertaisuuden tunteminen aiheuttivat stressiä jokapäiväisessä elämässäni. Toisaalta pystyn nyt varmistamaan, etten enää itse ole osa tiedostamatonta ongelmaa. Konventiot ja käyttäjälähtöinen suunnittelu olivat minulle jo entuudestaan tuttuja, mutta opin silti paljon pientä, joka auttaa minua jatkossa luomaan käyttäjää koskettavia sivuja.

Suunnitteluvaiheessa ei ilmennyt ongelmia. Tuntematon ala aiheutti päänvaivaa vain aivan projektin alussa ja siihen tutustuttuani olen nyt taas hieman viisaampi. Ohjaajalta ja ystäviltä saatu palaute auttoi ulkoasusuunnittelussa oikeaan suuntaan, ja pystyin ylittämään omat pessimistiset ennakko-odotukseni omista taidoistani. Sain myös asiakkaalta rakentavaa palautetta sivuston rakennetta koskien ja pystyin näin tarjoamaan myös häntä miellyttävän sivukokonaisuuden.

Työn ongelmat alkoivat vasta toteutusvaiheessa. Olin aliarvioinut uuden sisällönhallintajärjestelmän vaatiman ylimääräisen ajan koodia kirjoitettaessa. Tätä vaikeutti suuresti se, että koska SilverStripe on vielä tuore sisällönhallintajärjestelmä, ei sen dokumentaatiosta useinkaan löytynyt ratkaisua kohtaamilleni ongelmille. Oli äärimmäisen turhauttavaa ajautua ongelmaan, jonka syy oli täysin hämärän peitossa. Oliko syy minussa, koodissani, sisällönhallintajärjestelmässä vai palvelimella? Onnistuin kuitenkin ratkaisemaan suurimman osan haasteista, osan kiersin ja osaan sain apua ohjaajalta sekä opiskelutovereiltani. Huomasin myös, kuinka jatkuvasti sivustoa rakentaessani en pystynyt pitäytymään alkuperäisen suunnitelman rakenneratkaisuissa. Esimerkiksi toisen tason navigaatio muuttui kesken projektin pudotusvalikosta alisteiseksi staattiseksi

navigaatioksi. Varmasti toistuvampi sivustojen rakentaminen tulevaisuudessa opettaa minulle oikeasti parhaat rakenneratkaisut, joita pystyn heti suunnitteluvaiheessa käyttämään.

Myös käytettävyystestaukseen tutustuminen jäi laihaksi. En ehtinyt tehdä täysimittaista käytettävyystestausta edes suunnittelemassani mittakaavassa. Onneksi kuitenkin tiedän nyt, mitä käytettävyystestaus vaatii ja miten se tehdään, joten pystyn sen tekemään myöhemmin ennen sivuston julkistamista. Käytettävyystestaus on verkkosuunnittelun aliarvioitu ja pelätty osanen, joka on kuitenkin korvaamaton väline jokaisen verkkosuunnittelijan työssä, ja pyrin käyttämään sitä tulevaisuudessa useammin, projektin laajuudesta riippumatta.

# LÄHTEET

Built With. 2013a. SilverStripe Usage Statistics. Saatavissa: http://trends.builtwith.com/cms/SilverStripe [viitattu 26.3.2013].

Built With. 2013b. CMS Usage Statistics. Saatavissa: http://trends.builtwith.com/cms [viitattu 26.3.2013].

Grossman, J. 2006. The origins of Cross-Site Scripting (XSS). Saatavissa: http://jeremiahgrossman.blogspot.fi/2006/07/origins-of-cross-site-scripting-xss.html [viitattu 8.3.2013].

Heilmann, C. 2010. Web Security: Are You Part Of The Problem? Saatavissa: http://coding.smashingmagazine.com/2010/01/14/web-security-primer-are-you-part-of-the-problem/ [viitattu 26.2.2013].

Kaiser, P. 2012. 400 Worpress security vulnerabilities?! Saatavissa: http://wpmu.org/wordpress-security-vulnerabilities/ [viitattu 11.3.2013].

Järvinen, P. 2003. Salausmenetelmät. Jyväskylä: Docendo Finland.

Järvinen, P. 2004. Nörttisanakirja. Helsinki: WSOY.

Krug, S. 2006. Älä pakota minua ajattelemaan. Helsinki: Readme.fi.

Krug, S. 2010. Rocket Surgery Made Easy. Berkeley: New Riders.

Munroe, R. 2012. XKCD. Saatavissa: http://xkcd.com/936/ [viitattu 24.2.2013].

Packt Publishing. 2011. 2011 Open Source Awards. Saatavissa: http://www.packtpub.com/open-source-awards-home [viitattu 26.3.2013].

PHP. 2013. SQL Injection. Saatavissa: http://php.net/manual/en/security.database.sql-injection.php [viitattu 26.3.2013].

PHP Freak. 2008. PHP Security. Saatavissa: http://www.PHPfreaks.com/tutorial/PHP-security [viitattu 8.3.2013].

Quinn, L. & Gardner-Madras, H. 2010. Comparing Open Source Content Management Systems. Saatavissa: http://themobiusnetwork.com/pdfs/idealware_os_cms_2010_1.pdf [viitattu 12.11.2012].

Rouse, M. 2011. Definition: Advanced Encryption Standard (AES). Saatavissa: http://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard [viitattu 10.3.2013].

Salihefendic, A. 2009. How to prevent XSS attacks. Saatavissa: http://amix.dk/blog/post/19432 [viitattu 8.3.2013].

Stack Overflow. 2011. Security risks from user-submitted HTML. Saatavissa: http://stackoverflow.com/questions/7540700/security-risks-from-user-submitted-html [viitattu 26.2.2013].

Symantec. 2007. Symantec Internet Security Threat Report. Saatavissa: http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_exec_summary_internet_security_threat_report_xiii_04-2008.en-us.pdf [viitattu 8.3.2013].

Techie Corner. 2007. How to disable directory browsing using Apache web server. Saatavissa: http://www.techiecorner.com/106/how-to-disable-directory-browsing-using-htaccess-apache-web-server/ [viitattu 25.2.2013].

Tietoturvakoulu. 2013. Saatavissa: http://www.tietoturvakoulu.fi/index.html [viitattu 24.2.2013].

Tilastokeskus. 2012. Tietokoneen ja internet-yhteyden yleisyys kotitalouksissa 2/2001–11/2012. Saatavissa: http://www.stat.fi/til/kbar/2012/12/kbar_2012_12_2012-12-27_kuv_014_fi.html [viitattu 24.2.2013].

Tolvanen, P. 2009. Käsitesekamelskaa: julkaisujärjestelmä, CMS, portaali, sisällönhallintajärjestelmä. Saatavissa: http://vierityspalkki.fi/2009/11/03/kasitesekamelskaa-julkaisujarjestelma-cms-portaali-sisallonhallintajarjestelma/ [viitattu 14.11.2012].

Typografia: Kieltä vai visuaalisuutta? 2001. Toim. Brusila, Riitta. Porvoo: WSOY.

Viestintävirasto. 2010. Tietoturvakoulu. Saatavissa: http://www.viestintavirasto.fi/index/tietoturva/tietoturvakoulu.html [viitattu 24.2.2013].

Webhotellivertailu[2]. 2013. Webhotelli vertailu. Saatavissa http://www.webhotellivertailu2.fi/ [viitattu 14.11.2012].

WordPress. 2012. Saatavissa: http://en.wordpress.com/stats/ [viitattu 14.11.2012].

World Wide Web Consortium. 2001. URIs, URLs, and URNs: Clarifications and Recommendations 1.0. Saatavissa: http://www.w3.org/TR/uri-clarification/ [viitattu 20.3.2013].

24.11.2011. Ikääntyneiden netin käyttö kasvaa kohisten. Yle. Saatavissa: http://yle.fi/uutiset/ikaantyneiden_netin_kaytto_kasvaa_kohisten/545825 [viitattu 24.2.2013].

Kuvakaappaus Functional Safety Finlandin vanhoista sivuista, otettu 11.16. 013

FUNCTIONAL SAFETY FINLAND YHTEYSTIEDOT TOIMINNALLINEN TURVALLISUUS

FUNCTIONAL SAFETY FINLAND WWW.FSF.FI

**YRITYS:**

Functional Safety Finland on toiminnallisen turvallisuuden palveluita tuottava yritys.

Tarjoamme toiminnallisen turvalisuuden asiantuntijapalveluita sekä koulutusta yrityksille.

**Koulutustarjonta:**
- IEC 61508 yrityksen johdolle
- IEC 61508 mukainen suunnittelu, toteutus, testaus, käyttö, ylläpito sekä muutoksenhallinta
- Tarvittaessa koulutusta myös EN 62061, ISO 13849, IEC 61511 standardeista

**Riskien hallinta:**
- Prosessin tai koneen vaarallisten tapahtumien ja riskien tunnistaminen (useita eri menetelmiä)
- Riskien analysointi ja hallinta (useita eri menetelmiä)
- Turvallisuuden eheyden tasojen (SIL) tai suoritustasojen (PL) määrittäminen

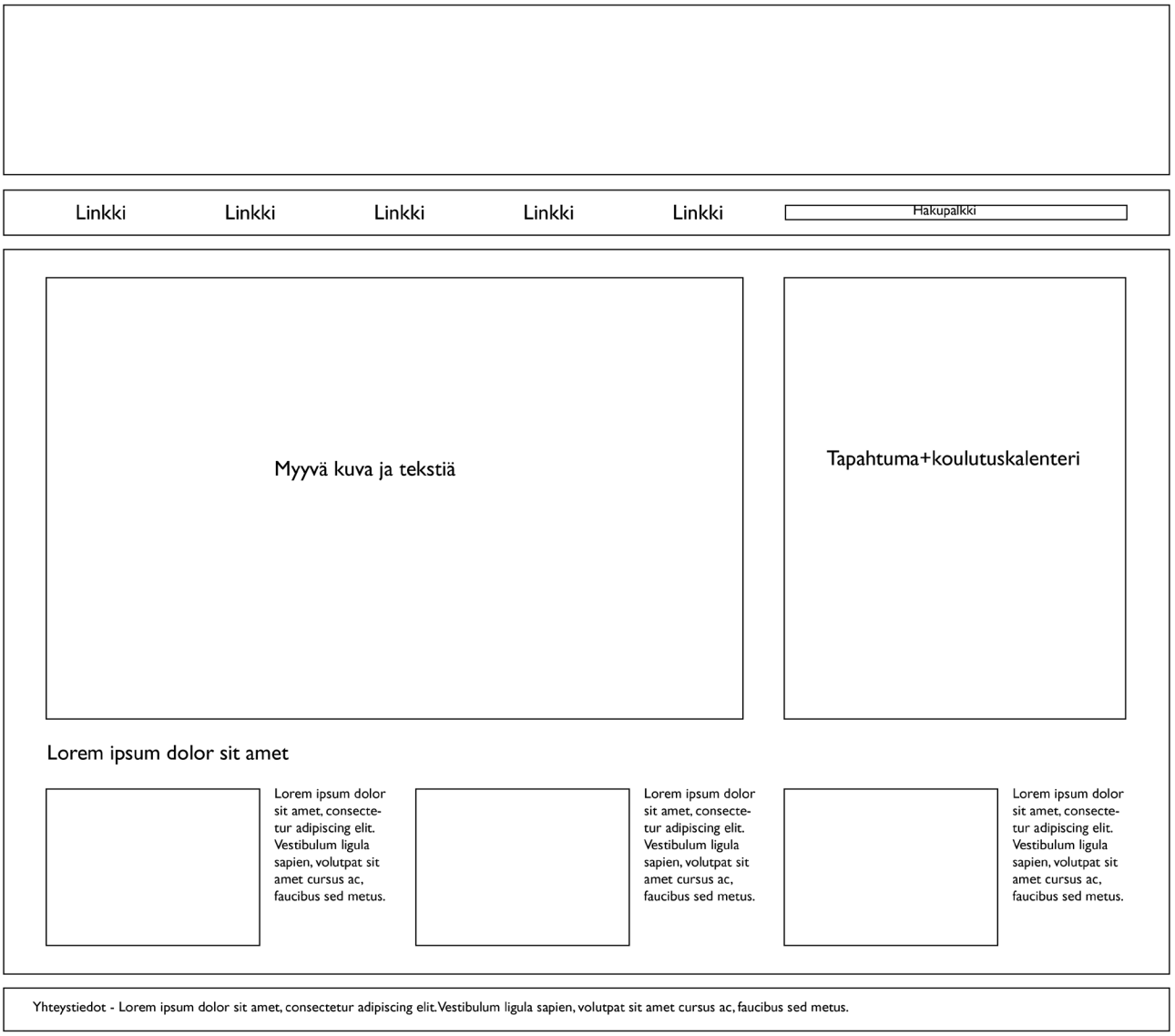
**Prosessit ja dokumentaatio:**
- TLJ:n elinkaaren suunnittelu (life cycle)
- Turvallisuusvaatimusmäärittelyt (SRS)
- Toiminnallisen turvallisuuden hallinnan suunnitelmat (FSMP)
- Todennus ja kelpoistus suunnitelmat (V&V)
- Muutoksen hallinnan määrittelyt
- TLJ:n dokumentaation määrittelyt

**Standardin mukaisuus:**

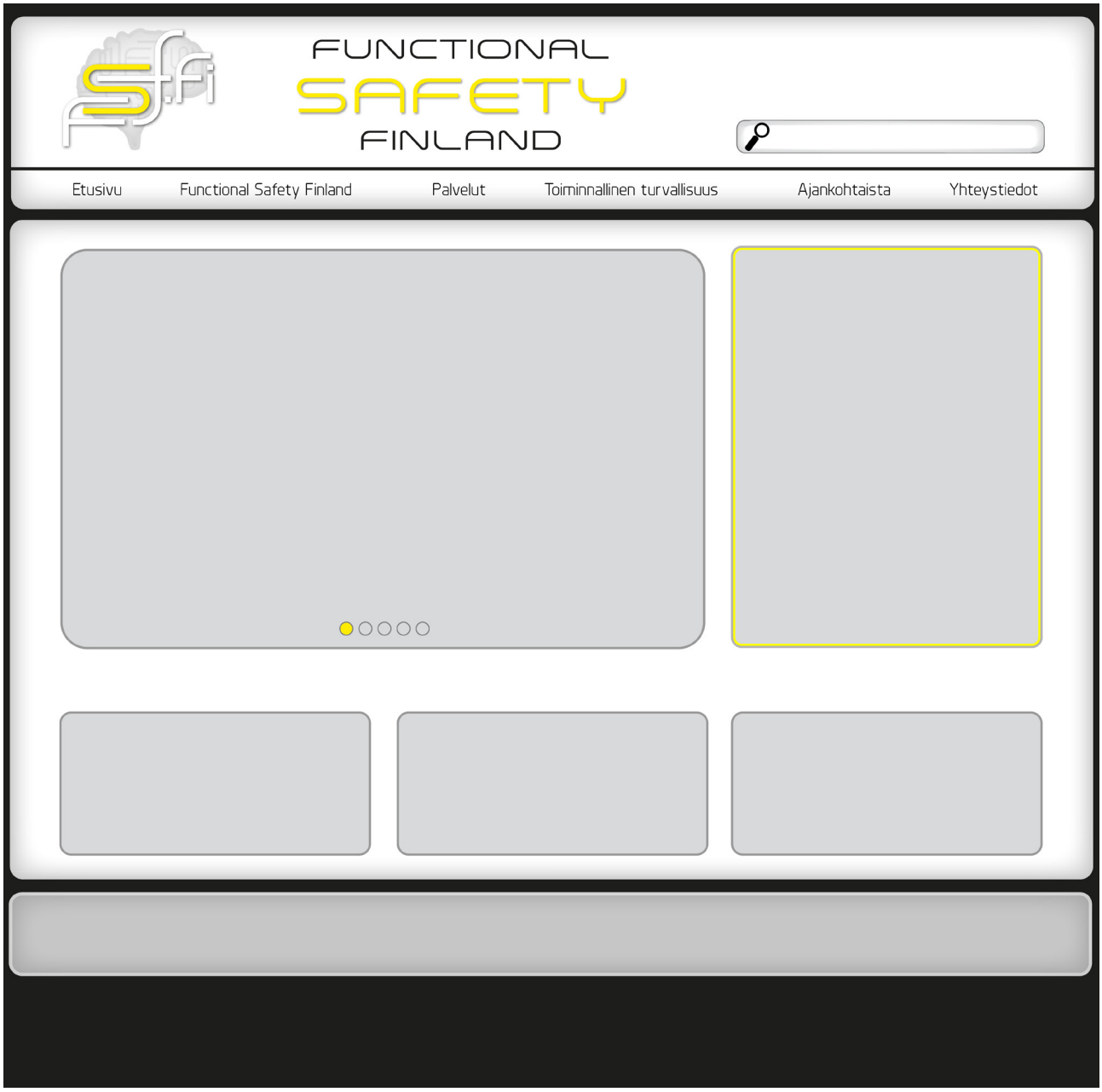
- Standardin mukaisuuden todistaminen (HW, SW, dokumentaatio, todennus, kelpoistus ja arviointi toimet)
- Toiminnallisen turvallisuuden auditoinnit ja arvioinnit

**Muuta:**
- Osallistuminen suunnittelupalavereihin toiminnallisen turvallisuuden asiantuntijana

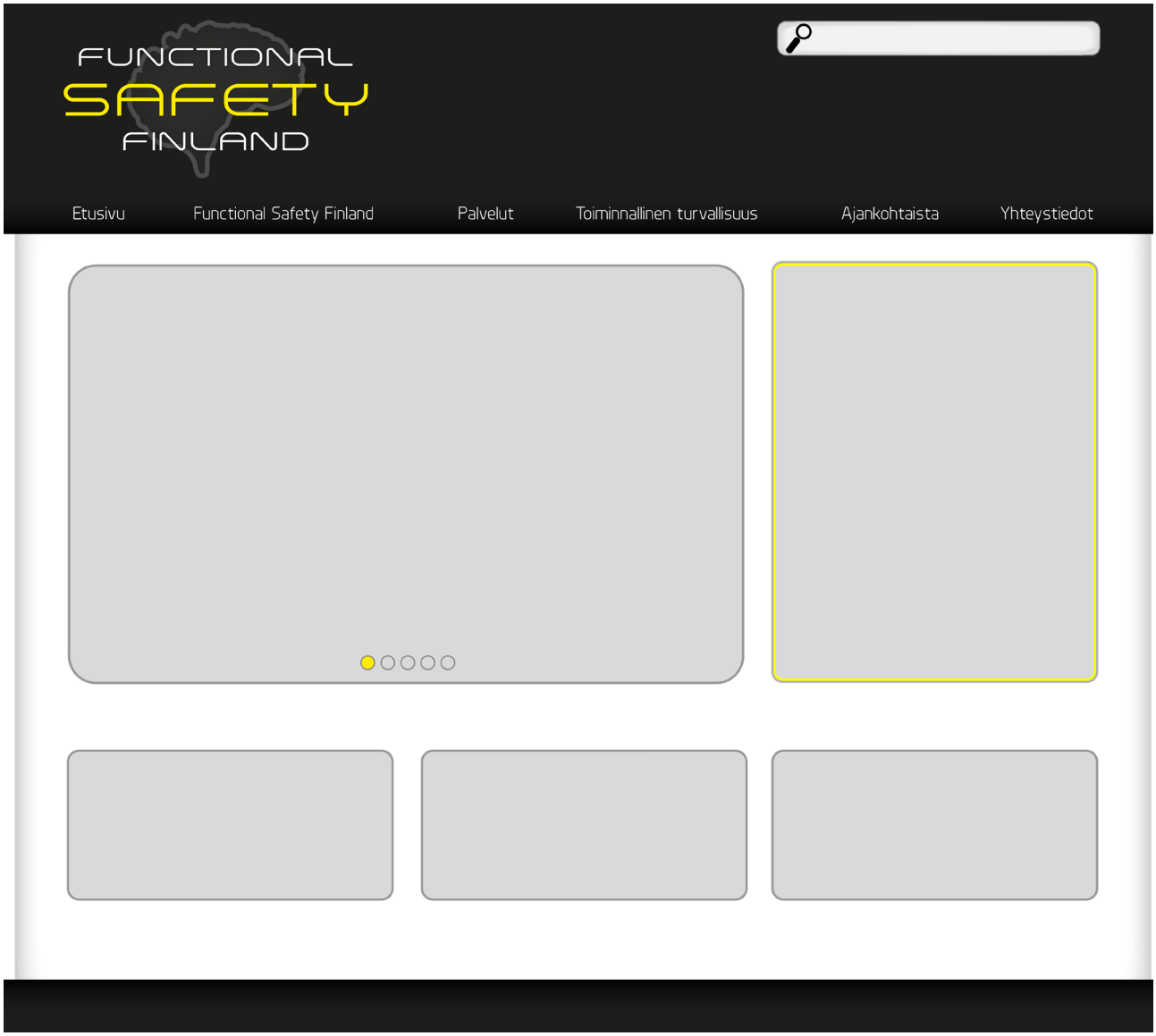
Yksi rautalankamalleista



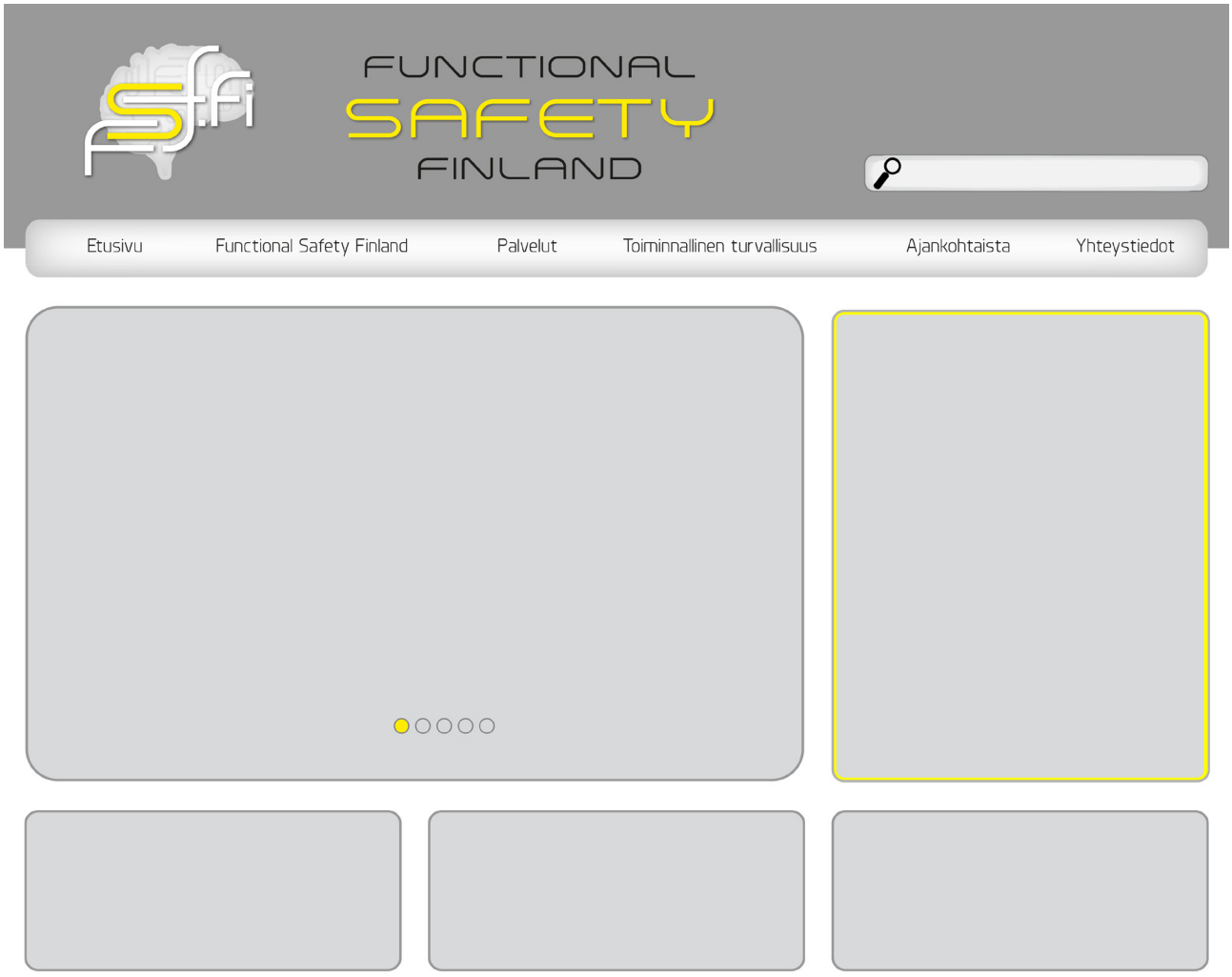
Ensimmäinen ulkoasuversio



Toinen ulkoasuversio



Kolmas ulkoasuversio



Neljäs ulkoasuversio, jossa näkyy myös kirjautumisikkuna pudotusvalikkona.

Viimeinen ulkoasuversio.